

**Universidad Carlos III de Madrid**

**Escuela Politécnica Superior**



**Ingeniería Técnica en Informática de  
Gestión**

**Proyecto Fin de Carrera**

**Diseño y Construcción de Bases de  
Datos Distribuidas Heterogéneas  
sobre Oracle Y SQL Server**

**Autor:** Laura Martínez Martín  
**Tutor:** M<sup>a</sup> Teresa Vicente Díez

**Octubre, 2010**



## Agradecimientos

Después de tanto tiempo y trabajo realizado, no podía olvidarme de agradecer:

El apoyo y el ánimo recibido por mis padres y mi hermano, sin el cual no habría acabado esta etapa de mi vida.

La paciencia demostrada en mis muchos momentos de desánimo y frustración, por Israel.

Y la comprensión y la posibilidad de poder compartir quejas y en ocasiones dudas con mi compañera y amiga Esther.

## Índice de Figuras

Ilustración 1. Clasificación de los SGBDD .....	19
Ilustración 2. Esquema de una fragmentación vertical.....	24
Ilustración 3. Esquema de una fragmentación horizontal.....	26
Ilustración 4. Esquema de una fragmentación vertical-horizontal. ....	27
Ilustración 5 Esquema de una fragmentación horizontal-vertical .....	27
Ilustración 6. Replicación de los datos .....	29
Ilustración 7. Procesamiento de una Consulta Distribuida.....	32
Ilustración 8 Ejemplo de un escenario de un SGBD Heterogéneo.....	38
Ilustración 9 Taxónomias de metodologías de la integración .....	42
Ilustración 10 Métodos de integración binarios .....	42
Ilustración 11 Métodos de Integración n-aria.....	43
Ilustración 12 Alternativas de Conformación Atómica.....	44
Ilustración 13 Estructura de un SGBDD heterogéneo.....	46
Ilustración 14 Pasos del procesamiento de consultas en Sistemas Multibase .....	47
Ilustración 15 Componentes de un SGBD heterogéneo [6] .....	49
Ilustración 16 Diagrama E/R correspondiente a una BD centralizada para la gestión de la Universidad Carlos III .....	59
Ilustración 17 Esquema relacional correspondiente a una BD centralizada para la gestión de la Universidad Carlos III .....	61
Ilustración 18 Esquema de la arquitectura del sistema.....	65
Ilustración 19 Abrir puerto de Firewall de Windows.....	71
Ilustración 20 Instalación de SQL Server 2008: Aprovisionamiento de cuentas.....	74
Ilustración 21 Oracle Universal Installer: Seleccionar producto a instalar .....	75
Ilustración 22 Oracle Universal Installer: Components Disponibles del Producto .....	76
Ilustración 23 Crear nuevo origen de datos.....	78
Ilustración 24 Configuración DSN de Microsoft SQL Server 1/3 .....	79
Ilustración 25 Configuración DSN de Microsoft SQL Server 2/3 .....	80
Ilustración 26 Configuración DSN de Microsoft SQL Server 3/3 .....	80
Ilustración 27 Prueba de conexión de XE .....	83
Ilustración 28 Opciones de proveedor - Oracle Provider for OLE Db.....	84
Ilustración 29 Nuevo Servidor Vinculado .....	84
Ilustración 30 Nuevo Servidor Vinculado - general.....	85
Ilustración 31 Nuevo Servidor Vinculado - seguridad.....	86
Ilustración 32 Prueba de conexión de red desde Sede de Leganés a sede de Getafe ....	87
Ilustración 33 Oracle Database 10g Express Edition – Asistente de Instalación 1/6.....	96
Ilustración 34 Oracle Database 10g Express Edition – Asistente de Instalación 2/6.....	97
Ilustración 35 Oracle Database 10g Express Edition – Asistente de Instalación 3/6.....	97
Ilustración 36 Oracle Database 10g Express Edition – Asistente de Instalación 4/6.....	98
Ilustración 37 Oracle Database 10g Express Edition – Asistente de Instalación 5/5.....	98
Ilustración 38 Oracle Database 10g Express Edition – Asistente de Instalación 6/6.....	99
Ilustración 39 Cómo acceder a la página de configuración y administración de Oracle	99
Ilustración 40 Validación en el acceso a la página de configuración y administración de Oracle.....	100
Ilustración 41 Ventana de administración de Oracle 10g Express Edition .....	101
Ilustración 42 Centro de Instalación de SQL Server .....	102
Ilustración 43 Programa de instalación de SQL Server 2008 1/21.....	103
Ilustración 44 Programa de instalación de SQL Server 2008 2/21.....	104

Ilustración 45 Programa de instalación de SQL Server 2008 3/21.....	105
Ilustración 46 Programa de instalación de SQL Server 2008 4/21.....	106
Ilustración 47 Programa de instalación de SQL Server 2008 5/21.....	107
Ilustración 48 Programa de instalación de SQL Server 2008 6/21.....	108
Ilustración 49 Programa de instalación de SQL Server 2008 7/21.....	109
Ilustración 50 Programa de instalación de SQL Server 2008 8/21.....	109
Ilustración 51 Usar la misma cuenta para todos los servicios de SQL Server 2008 ....	110
Ilustración 52 Programa de instalación de SQL Server 2008 9/21.....	110
Ilustración 53 Programa de instalación de SQL Server 2008 10/21.....	111
Ilustración 54 Programa de instalación de SQL Server 2008 11/21.....	112
Ilustración 55 Programa de instalación de SQL Server 2008 12/21.....	112
Ilustración 56 Programa de instalación de SQL Server 2008 13/21.....	113
Ilustración 57 Programa de instalación de SQL Server 2008 14/21.....	113
Ilustración 58 Programa de instalación de SQL Server 2008 15/21.....	114
Ilustración 59 Programa de instalación de SQL Server 2008 16/21.....	114
Ilustración 60 Programa de instalación de SQL Server 2008 17/21.....	115
Ilustración 61 Programa de instalación de SQL Server 2008 18/21.....	115
Ilustración 62 Programa de instalación de SQL Server 2008 19/21.....	116
Ilustración 63 Programa de instalación de SQL Server 2008 20/21.....	116
Ilustración 64 Programa de instalación de SQL Server 2008 21/21.....	117

## Índice de Tablas

Tabla 1 Comparación de las alternativas de replicación .....	30
Tabla 2 Resumen de los costes de personal conforme a las fases del proyecto .....	54
Tabla 3 Resumen del coste de los recursos personales del proyecto.....	54
Tabla 4 Resumen del coste de los recursos materiales del proyecto .....	55
Tabla 5 Cálculo del coste amortización de los recursos materiales .....	55
Tabla 6 Suma final de los costes materiales y personales .....	55
Tabla 7 Definición de identificadores de tipos de requisitos de usuario .....	56
Tabla 8 Definición de identificadores de tipos de requisitos de sistema.....	56
Tabla 9 Requisitos de usuario.....	57
Tabla 10 Requisitos de sistema .....	58
Tabla 11 Asignación inicial de tablas y nodos. ....	62
Tabla 12 Esquema de asignación .....	64
Tabla 13 Esquema de replicación.....	64
Tabla 14 Requerimientos hardware y software para instalar SQL Server .....	74
Tabla 15 Servicios de Oracle 10g Express Edition .....	101
Tabla 16 Accesos directos de Oracle 10g Express Edition .....	102

# Índice de contenidos

<b>1.</b>	<b>INTRODUCCIÓN .....</b>	<b>9</b>
1.1	CONTEXTO .....	9
1.2	OBJETIVOS .....	9
1.3	ESTRUCTURA DEL DOCUMENTO.....	10
<b>2</b>	<b>ESTADO DE LA CUESTIÓN .....</b>	<b>11</b>
2.1	INTRODUCCIÓN A LAS BASES DE DATOS DISTRIBUIDAS .....	11
2.1.1	<i>Definiciones.....</i>	12
2.1.2	<i>Características de las BDD.....</i>	13
2.1.3	<i>Ventajas de las BDD.....</i>	14
2.1.4	<i>Desventajas de las BDD.....</i>	15
2.2	SISTEMAS GESTORES DE BASES DE DATOS DISTRIBUIDAS (SGBDD) .....	16
2.2.1	<i>Características y arquitectura de los SGBDD.....</i>	16
2.2.2	<i>Clasificación de los SGBDD .....</i>	17
2.3	DISEÑO DE BDD .....	20
2.3.1	DEFINICIONES .....	20
2.3.2	<i>Alternativas de diseño .....</i>	21
2.3.3	<i>Fragmentación .....</i>	21
2.2.7	<i>Asignación de fragmentos.....</i>	28
2.2.8	<i>Alternativas de replicación.....</i>	28
2.3	CONTROL SEMÁNTICO DE LOS DATOS .....	30
2.4	PROCESAMIENTO DE CONSULTAS .....	31
2.4.1	<i>Definiciones.....</i>	31
2.4.2	<i>Descomposición de consultas globales.....</i>	32
2.4.3	<i>Localización de los datos.....</i>	33
2.5	CONTROL DE CONCURRENCIA EN BDD .....	34
2.6	CONTROL DE RECUPERACIÓN .....	36
2.7	BASES DE DATOS DISTRIBUIDAS HETEROGÉNEAS .....	37
2.7.1	<i>Tipos de heterogeneidades .....</i>	38
2.7.2	<i>Ventajas de las BBDD en sistemas heterogéneos.....</i>	39
2.7.3	<i>Problemas de BBDD en sistemas heterogéneos .....</i>	40
2.7.4	<i>Integración de las BBDD.....</i>	41
2.7.5	<i>Procesamiento de consultas en BBDD heterogéneas.....</i>	45
2.7.6	<i>Gestión de transacciones.....</i>	48
2.8	HERRAMIENTAS UTILIZADAS.....	50
2.8.1	<i>El SGBD Oracle .....</i>	50
2.8.2	<i>El SGBD SQLServer.....</i>	50
2.8.3	<i>Servidor Vinculado.....</i>	51
2.8.4	<i>Database Link.....</i>	51
<b>3</b>	<b>ESTUDIO DE LA VIABILIDAD Y PLANTEAMIENTO DEL PROYECTO .....</b>	<b>53</b>
3.1	DESCRIPCIÓN GENERAL DEL SISTEMA Y ALCANCE DEL PROYECTO .....	53
3.2	PLANIFICACIÓN TEMPORAL DEL PROYECTO Y ESTIMACIÓN DE COSTES .....	54
3.3	SUPUESTO PRÁCTICO.....	55
<b>4</b>	<b>ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>	<b>56</b>
4.1	ESPECIFICACIÓN DE REQUISITOS .....	56
4.1.1	<i>Requisitos hardware .....</i>	56
4.1.2	<i>Requisitos software.....</i>	56
4.1.3	<i>Requisitos de usuario y de sistema .....</i>	56
4.2	ESTRATEGIA DE DISEÑO DE LA BDD .....	58
	<i>Diseño centralizado .....</i>	58
4.2.1	<i>Diagrama E/R.....</i>	58
4.2.2	<i>Supuestos semánticos asociados al diagrama E/R.....</i>	59
4.2.3	<i>Supuestos semánticos no incluidos en el enunciado.....</i>	60
4.2.4	<i>Transformación al esquema relacional.....</i>	60

<i>Diseño de la distribución</i> .....	61
4.2.5 <i>Identificación de los nodos de distribución.</i> .....	61
4.2.6 <i>Asignación inicial de tablas y nodos.</i> .....	62
4.2.7 <i>Análisis de los criterios de fragmentación.</i> .....	62
4.2.8 <i>Asignación de fragmentos a las sedes.</i> .....	64
4.2.9 <i>Análisis de replicación</i> .....	64
4.2.10 <i>Esquema de replicación</i> .....	64
4.3 DISEÑO Y DESCRIPCIÓN DE LA ARQUITECTURA .....	64
4.3.1 <i>Sede Leganés</i> .....	65
4.3.2 <i>Sede Getafe</i> .....	67
<b>5 IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>69</b>
5.1 INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR ORACLE – SEDE LEGANÉS.....	69
5.1.1 <i>Instalación Oracle 10g Express Edition</i> .....	69
5.1.2 <i>Configuración del Servidor Oracle</i> .....	69
5.1.3 <i>Otras configuraciones</i> .....	71
5.2 INSTALACIÓN Y CONFIGURACIÓN DEL CLIENTE ORACLE Y DEL SERVIDOR SQLSERVER – SEDE GETAFE 72	72
5.2.1 <i>Instalación Oracle Database 10g Client Release 2</i> .....	72
5.2.2 <i>Instalación de SQL Server 2008</i> .....	72
5.2.3 <i>Instalación Oracle 10g Release 2 Oracle Data Access Components (ODAC)</i> .....	75
5.2.4 <i>Configuración del servidor SQL Server</i> .....	76
5.2.5 <i>Otras configuraciones</i> .....	77
5.3 CONFIGURACIÓN DEL CLIENTE Y EL SERVIDOR PARA CREAR EL ENTORNO DISTRIBUIDO E INTERCONECTAR LAS SEDES. ....	77
5.3.1 <i>Descripción y requisitos para conectar Oracle a SQL Server</i> .....	77
5.3.2 <i>Descripción y requisitos para conectar SQL Server a Oracle</i> .....	82
<b>6 EXPERIMENTACIÓN Y RESULTADOS .....</b>	<b>86</b>
6.1 PRUEBA DE CONECTIVIDAD DE RED ENTRE LAS SEDES .....	86
6.2 PRUEBA DEL SERVIDOR VINCULADO .....	87
6.3 PRUEBA DE ENLACES PÚBLICOS.....	88
6.4 PRUEBA DE LOS DISPARADORES QUE GARANTIZAN LA TRANSPARENCIA DE LA FRAGMENTACIÓN VERTICAL .....	88
6.5 CONSULTA Y RECUPERACIÓN DE LOS DATOS DE UNA TABLA FRAGMENTADA HORIZONTALMENTE.....	90
6.6 CONSULTA Y RECUPERACIÓN DE LOS DATOS DE UNA TABLA FRAGMENTADA VERTICAL Y HORIZONTALMENTE.....	90
<b>7 CONCLUSIONES .....</b>	<b>91</b>
<b>8 LÍNEAS FUTURAS.....</b>	<b>92</b>
<b>9 LISTA DE ACRÓNIMOS.....</b>	<b>93</b>
<b>10 BIBLIOGRAFÍA .....</b>	<b>94</b>
10.1 REFERENCIAS BIBLIOGRÁFICAS.....	94
10.2 REFERENCIAS EN INTERNET .....	95
<b>11 ANEXOS.....</b>	<b>96</b>
11.1 INSTALACIÓN DE ORACLE 10G EXPRESS EDITION .....	96
11.2 INSTALACIÓN DE MICROSOFT SQL SERVER 2008 .....	102



# 1. INTRODUCCIÓN

## 1.1 Contexto

Con los recientes avances tecnológicos en cuanto a comunicación y redes, hoy en día es cada vez más habitual encontrar sistemas descentralizados cuya información se encuentre disponible desde varias localizaciones geográficas.

Como parte clave de estos sistemas, las bases de datos también experimentan un cambio similar y cada vez es más frecuente que las organizaciones tengan sus sistemas de bases de datos ejecutándose en diversas plataformas, ya que algunas de las aplicaciones que utilizan están estrechamente ligadas a determinadas bases de datos y será deseable poder comunicarlás para que trabajen conjuntamente y poder así hablar de sistemas heterogéneos de bases de datos.

Sin embargo, a pesar de que el paradigma de las bases de datos distribuidas hizo su aparición hace ya años, el enfoque distribuido todavía no se ha extendido suficientemente desde el punto de vista de las herramientas comerciales para su implementación. Teniéndose en cuenta que los dos componentes básicos de los sistemas de bases de datos distribuidos (bases de datos y redes de ordenadores) son sistemas ampliamente desarrollados y cuyas ventajas son bien conocidas por las organizaciones, se puede conjeturar que su extensión comercial es un hecho inminente. Es de esperar que en un futuro próximo los principales sistemas gestores de bases de datos comerciales apuesten por este enfoque y que se produzca la consolidación de la resolución de los problemas que la distribución acarrea.

Mientras tanto, en este proyecto se plantea el estudio de las bases de datos distribuidas heterogéneas mediante el diseño y simulación de un sistema real que cumpla las características mencionadas anteriormente, explorando las posibilidades teóricas y prácticas existentes a través del empleo de algunos de los sistemas comerciales de bases de datos más frecuentes, analizando las posibles dificultades que se planteen.

## 1.2 Objetivos

Este proyecto surge por el deseo de ampliar el alcance de trabajos previos orientados a la investigación sobre sistemas homogéneos de bases de datos distribuidas, promoviendo en esta ocasión el estudio y desarrollo de los sistemas heterogéneos de base de datos.

El objetivo principal de este trabajo es investigar la problemática actual de las conexiones entre sistemas gestores de bases de datos de diferentes proveedores. Como objetivo secundario está el llevar a cabo un proyecto software y hardware, realizando el diseño y la implementación completa de un sistema de BBDD en diferentes sistemas gestores de base de datos y conocer así nuevas tecnologías. Son metas derivadas de los objetivos anteriores el diseño, la instalación, configuración y mantenimiento de dichos sistemas, de manera que se garantice al usuario final la transparencia del entorno con respecto a cuestiones internas de representación y gestión de los datos.

Este proyecto es llevado a cabo además por la necesidad de:

- ampliar las nociones de bases de datos distribuidas y bases de datos distribuidas heterogéneas en cuanto a diseño, control semántico de los datos, procesamiento de consultas, control de concurrencia y control de recuperación
- llevar a cabo un proyecto de base de datos en su fase de planificación, análisis, diseño, implementación y pruebas
- administrar una base de datos con las herramientas Oracle y SQL Server y conocer las limitaciones de estos gestores en sistemas distribuidos
- conocer e implementar los objetos que provee Oracle y SQL Server para comunicar bases de datos distribuidas heterogéneas
- encontrar formas de suplir las funciones que debe cumplir teóricamente un sistema gestor de base de datos distribuidas y que actualmente no desempeñan los actuales sistemas gestores de bases de datos comerciales

En el presente documento primeramente se introducen conceptos generales propios de las tecnologías de bases de datos, para posteriormente centrarse en los fundamentos de las bases de datos heterogéneas. Una vez definidos los conceptos necesarios para la comprensión de la cuestión a tratar, se realizará el estudio de la conexión entre diferentes Sistemas Gestores de Bases de Datos con algunas de las herramientas que actualmente existen en el mercado, tratando de localizar las características, ventajas, desventajas y usabilidad de cada una de ellas.

### ***1.3 Estructura del Documento***

El contenido de este documento se estructura en los siguientes capítulos:

- El capítulo 1 – Introducción.  
En este capítulo se comenta la motivación para la realización del proyecto, los objetivos que se pretenden llevar a cabo, y el punto actual de estructura del documento.
- El capítulo 2 – Estado de la cuestión.  
En este capítulo se describen ciertos conceptos teóricos relacionados con el proyecto:
  - Relativos a las Bases de Datos Distribuidas (BDD), sus características, ventajas y desventajas, sistemas gestores de Bases de Datos, diseño de Bases de Datos Distribuidas, control semántico de los datos, procesamiento de consultas, control de concurrencia, control de recuperación.
  - Relativos a las BDD heterogéneas, tipos, ventajas, desventajas, procesamiento de consultas y gestión de transacciones.
- El capítulo 3 - Estudio de la viabilidad y planteamiento del proyecto.  
En este capítulo se realiza la descripción general del sistema, la descripción del entorno, y el alcance del proyecto. Se incluye también el análisis de viabilidad, planificación temporal y estimación de costes .
- El capítulo 4 - Análisis y diseño del sistema.  
En este capítulo se especifica los requisitos hardware, software, de usuario y de sistema, la identificación de los nodos de distribución y la asignación de los fragmentos en los mismos.

- El capítulo 5 - Implementación del sistema.  
En este capítulo se especifican aspectos relacionados con la implementación llevada a cabo para el desarrollo del proyecto.
- El capítulo 6 - Experimentación y resultados.  
En este capítulo se documentan los resultados de la ejecución de las pruebas realizadas durante la conexión desde las diferentes sedes del entorno distribuido y las acciones correctivas, si procede.
- El capítulo 7 – Conclusiones.  
En este capítulo se comentan las conclusiones alcanzadas de lo aprendido tras la finalización del proyecto.
- El capítulo 8 - Líneas futuras.  
En este capítulo se comentan las posibles ampliaciones y mejoras que se pueden realizar en un futuro sobre la aplicación.
- El capítulo 9 - Lista de acrónimos.  
En este capítulo se incluyen todos los acrónimos que se han mencionado en algún momento en esta memoria, con su significado.
- El capítulo 10 – Bibliografía.  
Lista de la bibliografía utilizada tanto en formato libro como páginas Web para la realización de este documento.
- El capítulo 11 – Anexos.  
En este capítulo se incluyen los anexos con información relevante del proyecto pero que se ha creído necesaria excluir de la línea principal de la memoria.

## 2 Estado de la cuestión

En este capítulo se introducen ciertos conceptos necesarios para la comprensión de este trabajo: conceptos acerca de la distribución de base de datos, de las diferentes arquitecturas de sus sistemas y de sus posibles diseños, entre otros. Del mismo modo, se da un enfoque substancial a las bases de datos distribuidas heterogéneas.

### 2.1 Introducción a las Bases de Datos Distribuidas

Las **Bases de Datos Distribuidas** (BDD) son la unión de dos conceptos que, en principio, pueden parecer opuestos: los **Sistemas Gestores de Bases de Datos** (SGBD), y por otro lado, los **sistemas en red**. La primera parte, los SGDB, son sistemas, con sus correspondientes aplicaciones y archivos, de gestión y almacenamiento de datos que están disponibles en un solo lugar. Bajo estos sistemas se implantan programas independientes entre sí, los cuales son totalmente indiferentes a cambios en la organización lógica y física de los datos y viceversa. La tecnología de redes de computadores, por otra parte, promueve un modo de trabajo que va en contra de todo esfuerzo de centralización.

A simple vista resulta difícil entender cómo estos dos enfoques, que se oponen, pueden ser sintetizados para producir una tecnología que es mucho más poderosa que cualquiera de los dos enfoques por separado.

La clave para entender esto es darse cuenta de que en la tecnología de las BDD el objetivo principal no es la centralización sino más bien la integración. La idea consiste en lograr la integración y distribución y que ambos enfoques no pierdan su naturaleza esencial.

### **2.1.1 Definiciones**

#### **Base de Datos (BD)**

Una base de datos es un conjunto de datos relacionados entre sí. Por datos se entienden hechos conocidos que pueden registrarse y que tienen un significado implícito. Por ejemplo, considérense los nombres y números de teléfonos de personas. Tal vez estos datos estén registrados en una libreta de direcciones indexada, o quizá en un disquete, utilizando un ordenador y un software. Se trata de un conjunto de datos relacionados entre sí y que tienen un significado implícito; por tanto, constituyen una base de datos.

La definición anterior es muy general pero la aceptación común del término base de datos suele ser más restringida. Una BD tiene las siguientes propiedades implícitas [1]:

- Una BD representa algún aspecto del mundo real, en ocasiones llamado universo del discurso. Las modificaciones del universo del discurso se reflejan en la BD.
- Una BD es un conjunto de datos lógicamente coherente, con cierto significado incoherente. Una colección aleatoria de datos no puede considerarse propiamente una BD.
- Toda BD se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

En otras palabras, una BD tiene una fuente de la cual se derivan los datos, cierto grado de interacción con los acontecimientos del mundo real y un público que está activamente interesado en el contenido de la BD.

Las BBDD pueden ser de cualquier tamaño y tener diversos grados de complejidad pero deben organizarse y controlarse para que los usuarios puedan buscar, obtener y actualizar los datos cuando sea necesario.

#### **Sistema de Gestión de BD (SGBD)**

Un SGBD es un conjunto de programas que permite a los usuarios crear y mantener una BD. Por tanto, el SGBD es un sistema de software de propósito general que facilita el proceso de definir, construir y manipular BBDD para diversas aplicaciones. Para definir una BD hay que especificar los tipos de datos, las estructuras y las restricciones de los datos que se almacenarán en ella. Construir una BD es el proceso de guardar los datos mismos en algún medio de almacenamiento controlado por el SGBD. En la manipulación de una BD intervienen funciones como consultar la BD para obtener datos específicos, actualizar la BD para reflejar cambios en el minimundo y generar informes a partir de los datos.

No es necesario un software de SGBD de propósito general para implementar una BD computerizada. Se podría escribir un conjunto de programas propio para crear y mantener la BD [1].

### **Sistema de Computación Distribuido.**

Es un número de elementos de procesamiento (no necesariamente homogéneos) que están interconectados por una red de computadoras y que cooperan para realizar las tareas que tienen asignadas [1].

### **Red de Computadoras.**

Una red de computadoras, también llamada red de ordenadores o red informática, es un conjunto de equipos (computadoras y/o dispositivos) conectados por medio de cables, señales, ondas o cualquier otro método de transporte de datos, que comparten información (archivos), recursos (CD-ROM, impresoras, etc.) y servicios (acceso a Internet, e-mail, Chat, juegos), etc.

Una red de comunicaciones es un conjunto de medios técnicos que permiten la comunicación a distancia entre equipos autónomos (no jerárquica *-master/slave-*). Normalmente se trata de transmitir datos, audio y vídeo a través de diversos medios (aire, vacío, cable de cobre, fibra óptica, etc.) [1].

### **Bases de Datos Distribuidas.**

Una Base de Datos Distribuida o BDD es una colección de datos integrados lógicamente en una BD, pero que físicamente pueden ser procesados y almacenados en varios nodos distribuidos sobre una red de ordenadores [2].

### **Sistema de Gestión de Bases de Datos Distribuido (SGBDD).**

Un SGBDD es un producto software que permite el manejo de Bases de Datos Distribuidas (BDD) y hace la distribución transparente a los usuarios [2].

### **Sistema de Bases de Datos Distribuidas (SBDD)**

Un SBDD es la unión entre un SGBDD y una BDD. Es decir, *es un sistema en el cual múltiples sitios de BD están conectados en una red de computadoras de tal forma que, un usuario en cualquier sitio puede acceder a los datos en cualquier parte de la red exactamente como si los datos estuvieran siendo accedidos de forma local* [4].

## **2.1.2 Características de las BDD**

Podemos caracterizar las BDD en varios aspectos [2]:

- Los **datos deben estar físicamente en más de un ordenador**, es decir, los datos se encuentran almacenados en distintas sedes.
- Las **sedes deben estar interconectadas mediante una red de computadoras** (cada sede será un nodo de la red). Para realizar el diseño no se tendrá en cuenta la topología, tipo y rendimiento de la red, aunque estas propiedades tengan relevancia en el buen funcionamiento del sistema.
- Los **datos han de estar lógicamente integrados en una única estructura** o esquema lógico global común.
- Los **usuarios han de tener acceso (recuperación y actualización) a los datos** pertenecientes a la BDD, ya residan éstos en la misma sede (acceso local) o en otra sede (acceso remoto).

- Cada nodo o emplazamiento facilita un entorno para la ejecución de **transacciones tanto locales como globales**.

- En una **única operación**, tanto de consulta como de actualización, se puede **acceder a datos que se encuentran en más de una sede sin que el usuario sepa la distribución de los mismos** en las distintas sedes. Es decir, que la distribución de la información sea transparente para el usuario.

### 2.1.3 Ventajas de las BDD

Entre las ventajas potenciales de los sistemas de BDD se encuentran las siguientes [1], [3]:

- **La naturaleza distribuida de algunas aplicaciones de bases de datos.** Muchas de estas aplicaciones están distribuidas naturalmente en diferentes lugares. Por ejemplo, una compañía puede tener oficinas en varias ciudades, o un banco puede tener múltiples sucursales. Es natural que las bases de datos empleadas en tales aplicaciones estén distribuidas en esos lugares. Muchos usuarios locales tienen acceso exclusivamente a los datos que están en el lugar, pero otros usuarios globales pueden requerir acceso ocasional a los datos almacenados en varios de estos emplazamientos.

- **Mayor fiabilidad y disponibilidad.** La fiabilidad se define a grandes rasgos como la probabilidad de que un sistema esté en funciones en un momento determinado. La disponibilidad es la probabilidad de que el sistema esté disponible continuamente durante un intervalo de tiempo. Cuando los datos y el software del sistema gestor de base de datos están distribuidos en varios sitios, un sitio puede fallar mientras que los demás siguen operando. Sólo los datos y el software que residen en el sitio que falló están inaccesibles. Esto mejora tanto la fiabilidad como la disponibilidad. En un sistema centralizado, el fallo de un solo sitio hace que el sistema completo deje de estar disponible para todos los usuarios.

- **Posibilidad de compartir los datos al tiempo que se mantiene un cierto grado de control local.** En algunos tipos de Sistemas de Bases de Datos Distribuidas (SBDD), es posible controlar los datos y el software localmente en cada sitio. Sin embargo, los usuarios de otros sitios remotos pueden tener acceso a ciertos datos a través del software del SGBDD. Esto hace viable poder compartir de forma controlada los datos en todo el sistema distribuido.

- **Mejor rendimiento.** Cuando una base de datos grande está distribuida en múltiples sitios, hay bases de datos más pequeñas en cada uno de éstos. En consecuencia, las consultas locales y las transacciones que tienen acceso a datos de un solo sitio tienen un mejor rendimiento porque las bases de datos locales son más pequeñas. Además, cada sitio tiene un menor número de transacciones en ejecución que si todas las transacciones se enviaran a una sola base de datos centralizada. En el caso de transacciones que impliquen acceso a más de un sitio, el procesamiento en los diferentes sitios puede efectuarse en paralelo, reduciéndose así el tiempo de respuesta.

- **Plataforma autónoma.** Dado que las aplicaciones y bases de datos no residen en las mismas máquinas, no hay ninguna razón particular para residir en el mismo tipo de máquina. Esta libertad permite a los administradores de bases de datos, desarrolladores y usuarios de escritorio elegir sus plataformas sin arriesgar cambios de funcionalidad en el

motor de base de datos.

- **Localización transparente.** Esto significa que ni las aplicaciones ni los usuarios necesitan preocuparse de dónde residen los datos o cómo se distribuyen. Al ser protegidos por estas características específicas la usabilidad de una base de datos aumenta ya que los desarrolladores y los usuarios no necesitan tener en cuenta detalles tales como cadenas de conexión. Por otra parte, los datos pueden ser trasladados de una instancia de base de datos a otra con un impacto mínimo sobre los usuarios y aplicaciones.

- **Autonomía del sitio.** Las BDD permiten varias localizaciones para compartir sus datos sin ceder el control administrativo. Si la sede de una instancia de base de datos contiene información especialmente sensible o tiene requerimientos de alta disponibilidad, todavía puede compartir datos sin comprometer su seguridad o la disponibilidad. Además, un sitio determinado en un entorno de BDD puede seguir sus propios procedimientos administrativos y actualización de rutas, dentro de lo razonable.

- **Seguridad mejorada.** Los componentes de la arquitectura distribuida son completamente independientes unos de otros, lo que significa que cada sitio puede mantenerse de forma independiente. Se pueden compartir los datos sin compartir cuentas y contraseñas. Cada sitio puede tener sus propios administradores y sus propios conjuntos de cuentas, y los datos privados se pueden mantener en privado. También se puede configurar un entorno distribuido para proporcionar seguridad en el sentido de la supervivencia, es decir, puede mantener dos o más versiones del esquema completo de la BD replicados en diferentes máquinas en diferentes lugares. En BDD no hay ninguna razón para que los desarrolladores o usuarios finales tengan su cuenta en un servidor de base de datos, porque todos los accesos de base de datos se producen a través de APIs (*Application Programming Interfaces*) de red. De este modo, se minimiza la exposición de dicho servidor de base de datos a posibles intrusiones maliciosas o a usuarios poco cuidadosos.

#### 2.1.4 Desventajas de las BDD

En contra, la distribución produce un aumento en la complejidad del diseño y en la implementación de nuevos sistemas [3].

- **Seguridad.** Esta característica también aparece en el apartado de ventajas, dado que tiene dos caras, debido a que puede ser difícil saber y controlar quién está accediendo a una base de datos mediante un enlace de BD. Las cuentas a las cuales se conecta un enlace de BD no debe dar más derechos de acceso que los absolutamente necesarios.

En un entorno de replicación avanzada, los problemas de seguridad pueden llegar a ser complicados porque la comunidad de usuarios puede ser el conjunto de todos los usuarios de todas las bases de datos que participan en la distribución. El mantenimiento de las cuentas en sí mismas puede convertirse en un trabajo de tiempo completo. Habrá que decidir si las operaciones de replicación deben ser realizadas en sitios remotos por el usuario original o por una cuenta de replicación genérica.

- **Coherencia de los datos.** Si existen varias réplicas una de las tareas de mayor esfuerzo será garantizar la coherencia de datos. Esta responsabilidad es compartida entre

los diseñadores, desarrolladores y administradores de bases de datos.

- **Administración de transacciones.** Si se necesita hacer un número de operaciones, pero tenemos la información en varios sitios replicada, debemos entender que el conjunto de operaciones debe ser realizado para cada sitio replicado. Si una solución posible es realizar la actualización por la noche, hora española, ¿qué ocurre con las sedes que están fuera de España y tienen un horario diferente y que quizá corresponda con su horario normal de trabajo? La cuestión es que las actualizaciones en un entorno replicado deben ser cuidadosamente coordinadas con todos los sitios a fin de evitar conflictos masivos y congestiones.

En la carga inicial y la distribución de datos entre los sitios también se requiere coordinación. Por ejemplo, puede ser interesante bloquear a los usuarios de todos los nodos hasta que se pueda garantizar que los datos son idénticos en todas las sedes.

- **Seguimiento.** La carga de trabajo de un DBA en un entorno distribuido puede ser considerable. El DBA debe controlar objetos tales como los registros de *snapshots*, las colas de trabajos, colas de transacciones, y las colas de errores. Si los problemas en un entorno distribuido se dejan sin resolver, puede ser tan difícil corregirlos que incluso sea más fácil recargar los datos desde el principio que tratar de solucionar errores específicos.

- **Recuperación.** Si una base de datos que forma parte de un entorno distribuido falla, el proceso de recuperación no sólo debe garantizar la restauración completa de los datos locales, sino también la restauración de los datos distribuidos, tales como las transacciones e instancias suspendidas.

- **Rendimiento.** Varios factores pueden afectar el rendimiento en una base de datos distribuida. Si la aplicación referencia datos desde un enlace de base de datos, el rendimiento de la red tendrá una influencia directa en el funcionamiento de la BDD. Los componentes de replicación que utilizan técnicas de almacenamiento y reenvío, tales como instantáneas y maestros de replicación, también tendrán su coste en el rendimiento global del sistema. El almacenamiento y la transmisión de operaciones impacta en el rendimiento general del sistema, y se debe tener en cuenta lo que esto significa cuando se especifican los requisitos de hardware.

## ***2.2 Sistemas Gestores de Bases de Datos Distribuidas (SGBDD)***

### **2.2.1 Características y arquitectura de los SGBDD**

Los SGBDD son un producto software que permiten la gestión de la base de datos y posibilitan una distribución transparente a los usuarios.

El SGBDD debe contar con funciones de un SGBD (Véase la definición de SGBD en el apartado 2.1.1) centralizado y además con funciones tales como:

- Capacidad de acceder a sitios remotos y transmitir consultas y datos entre diversos sitios a través de una red de ordenadores.
- Capacidad de rastrear la traza de distribución y de replicación de los datos en el



catálogo de SGBDD.

- Capacidad de elaborar estrategias de ejecución para consultas y transacciones con acceso a datos situados en nodos diferentes.
- Capacidad de mantener la consistencia en las réplicas de un elemento de información.
- Capacidad de decidir cuál de las copias de un elemento de información será accedida.
- Capacidad de recuperación ante caídas de sitios individuales y fallos de un enlace de comunicación.

El SGBDD debe contar con los siguientes componentes para poder ofrecer estas funcionalidades descritas:

- **Procesador de datos locales** que se encarga de la gestión local de los datos de forma parecida al software de un SGBD centralizado, por lo que además de ejecutar transacciones locales se encarga de la concurrencia y la recuperación ante fallos a nivel local.

- **Diccionario o directorio global** donde se guardará información acerca de dónde y cómo se almacenan los datos, el modo de acceso y otras características físicas. En resumen, contiene las especificaciones necesarias para pasar de la representación externa o esquema externo de los datos a la representación interna de los mismos.

- **Procesador de aplicaciones distribuidas** que es el responsable de las funciones distribuidas. Accede a la información sobre la ubicación de los datos, que se encuentra en el diccionario, y se ocupa de procesar todas las peticiones que involucran más de una sede para generar un plan de ejecución distribuido. Es el elemento diferenciador en los sistemas distribuidos, dada una operación se encargará de repartir el trabajo a los distintos procesadores locales que intervienen en dicha operación.

- **Software y red de comunicaciones.** No forma parte estrictamente del SGBDD sino que provee al procesador de aplicaciones distribuidas primitivas y servicios de comunicaciones para que éste lleve a cabo su labor.

## 2.2.2 Clasificación de los SGBDD

Esta clasificación se realiza en base a tres características: la distribución, la autonomía y la heterogeneidad del sistema local.

### Distribución

Los datos pueden ser distribuidos entre múltiples bases de datos. Estas bases de datos pueden estar almacenadas en un único sistema informático o en varios, distribuidos geográficamente, pero interconectados por un sistema de comunicación. Los datos pueden ser distribuidos entre múltiples bases de datos de diferentes maneras. Estos incluyen, en términos relacionales, particiones de base de datos verticales y horizontales.

Además pueden mantenerse varias copias de todo o parte de los datos. Estas copias no tienen por qué tener una estructura idéntica.

Algunos de los beneficios de la distribución de los datos son bien conocidos, tales como una mayor disponibilidad y fiabilidad de la información, así como una mejora en los tiempos de acceso [6].

### **Autonomía**

Se refiere a la distribución del control, no de los datos. Indica el grado en el que los SGBDs pueden actuar independientemente y es función de factores tales como el nivel de intercambio de información entre los componentes del sistema, la ejecución independiente de transacciones, la tolerancia a modificaciones respectivas en los datos, etc.

Para lograr la independencia entre bases de datos que componen un sistema distribuido, es necesario considerar los tipos de autonomía descritos por [6]:

- **Autonomía de diseño:** La capacidad de que cada nodo de la base de datos decida los aspectos relacionados con su diseño. Es decir, las personas involucradas son libres de decidir cualquier particularidad e incluso decidir qué SGBD utilizar. Los aspectos a considerar en la autonomía de diseño son:

- El universo de datos relevante para el sistema
- La representación (modelo de datos, lenguaje de consultas) y el nombrado de los datos
- La conceptualización o interpretación semántica de los datos (heterogeneidad semántica)
- Restricciones usadas para administrar los datos
- La funcionalidad del sistema
- La asociación y compartición con otros sistemas
- La implementación (por ejemplo: registros y estructuras de archivos).

- **Autonomía de comunicación:** es la habilidad de que un nodo decida comunicarse o no con otro componente de una misma federación

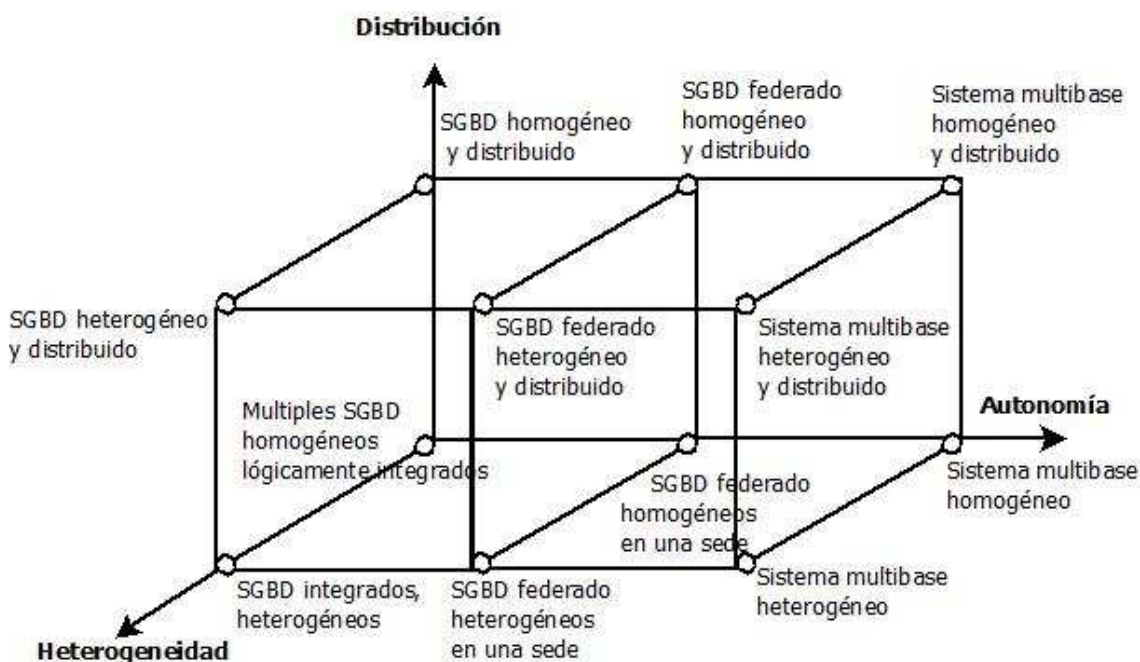
- **Autonomía de ejecución:** es la habilidad de un nodo para ejecutar operaciones locales sin la interferencia de operaciones externas, en el orden en que el nodo lo decida.

- **Autonomía de asociación:** cada nodo cuánto y cuándo puede compartir su funcionalidad y recursos con otros componentes, inclusive la capacidad de asociarse o retirarse de una o más federaciones.

## Heterogeneidad

La heterogeneidad puede referirse a diferentes aspectos, desde diferencias hardware o en los protocolos de red, hasta variaciones en los SGBDs propiamente (modelos de datos, lenguaje de manipulación de datos, protocolos de gestión de transacciones,...) [6].

En la Ilustración 1 se pueden observar las diferentes alternativas en las arquitecturas de los SGBDD según su grado de autonomía, distribución y heterogeneidad.



**Ilustración 1. Clasificación de los SGBDD**

A continuación se explican las diferentes clasificaciones a partir de las mencionadas alternativas en las arquitecturas de los SGBDD [4].

- La clase con múltiples SGBDs homogéneos lógicamente integrados. Si no hay distribución ni heterogeneidad el sistema es un conjunto de múltiples SGBDs que están lógicamente integrados. No hay sistemas de este tipo pero es una solución especialmente adecuada para entornos multiprocesador.
- Al introducir heterogeneidad (Clase con SGBDs heterogéneos integrados y SGBDs distribuidos heterogéneos) resultan múltiples SGBDs pero con una visión integral para el usuario. Es típico de sistemas diseñados para integrar el acceso a través de la red a bases de datos jerárquicas o relacionales que residen completamente en un nodo.
- Al introducir distribución de los datos (SGBD homogéneo y distribuido), el caso se generaliza. Esto es lo más interesante, la distribución de los datos más allá de la heterogeneidad y la autonomía de los SGBDs.
- En un grado intermedio de autonomía están los denominados sistemas federados (SGBD federado heterogéneo en una sede, SGBD federado heterogéneo y distribuido y

SGBD federado homogéneo y distribuido), de forma que existe una cooperación entre los SGBDs implicados en la ejecución de las solicitudes de usuarios a los datos de la BDD.

- Si la autonomía es total (sistema multibase heterogéneo, sistema multibase heterogéneo y distribuido y sistema multibase homogéneo y distribuido) se obtienen sistemas de bases de datos múltiples (SMBD).

- Sin la heterogeneidad (sistemas multibase homogéneos y distribuidos) o distribución (multibase heterogéneo) un sistema multibase es una colección de BDs autónomas interconectadas.

- Si la BD está distribuida (sistema multibase heterogéneo y distribuido, sistema multibase homogéneo distribuido) resulta un sistema gestor multibase distribuido que requiere una gestión bastante diferente de la de un SGBDD. Esto es porque la autonomía es el factor determinante en el diseño de un sistema gestor multibase.

## **2.3 Diseño de BDD**

En el diseño de las BDD, una cuestión clave es la distribución de los datos. El encargado de esta tarea es el DBA, el cual establece en tiempo de diseño el modelo de distribución de los datos. Esta definición es progresiva, de acuerdo a la evolución de la BD y al modo en que el SGBD resuelve:

- La distribución de las funciones
- La sincronización de las operaciones
- La descomposición de las consultas

### **2.3.1 Definiciones**

#### **Algebra relacional**

*Es un conjunto de operaciones que describen paso a paso cómo computar una respuesta sobre las relaciones, tal y como éstas son definidas en el modelo relacional. Denominada de tipo procedimental.*

*Describe el aspecto de la manipulación de datos. Estas operaciones se usan como una representación intermedia de una consulta a una base de datos y, debido a sus propiedades algebraicas, sirven para obtener una versión más optimizada y eficiente de dicha consulta. [5]*

Se van a explicar algunos operadores de algebra relacional, que nos van a ser necesarios para entender las reglas de fragmentación (Véase apartado 2.2.6.5).

#### **JOIN**

La sentencia *join* en SQL permite combinar registros de dos o más tablas en una base de datos relacional. Para hacer un *join* es necesario que exista un campo que relacione los datos de cada tabla.

#### **SEMI-JOIN**

También conocida como semi-combinación. Opera de la misma manera que un *join*, pero la relación resultante sólo tiene los atributos de la primera tabla.

## UNION

La sentencia *union* en SQL permite combinar los resultados de dos consultas SQL en una tabla única de todas las filas coincidentes. Las dos consultas deben tener el mismo número de columnas y tipos de datos compatibles para unir. Cualquier registro duplicado se elimina automáticamente si se usa *UNION ALL*.

### 2.3.2. Alternativas de diseño

En todo proceso de diseño hay dos aproximaciones básicas [2]:

- la ascendente o Top-Down.
- la descendente o Bottom-Up.

El **diseño ascendente** se utiliza, cuando se crea un BDD a partir de varias BBDD locales existentes. Se parte de varios esquemas lógicos locales (ELL) con ubicaciones diferentes que se integran en un único esquema lógico global (ELG).

El **diseño descendente** parte de un ELG y construye varios ELL definidos a partir de esquemas de fragmentación, asignación y replicación que determinan la distribución de los datos en los distintos nodos de la red.

### 2.3.3. Fragmentación

La fragmentación es un conjunto de técnicas para dividir la BD en unidades lógicas, llamadas fragmentos, cuyo almacenamiento puede asignarse a los diversos sitios. Estas técnicas se utilizan durante el proceso de diseño de BDD. La información concerniente a la fragmentación de los datos se almacena en un catalogo global del sistema al que tiene acceso el software cliente cuando es necesario [1].

*La fragmentación es un concepto que surge en los modelos que incluyen algún tipo de particionado y tiene implicaciones físicas y lógicas [4].*

- *Fragmento lógico: Bloque de datos localizado en algún SGBDD que puede ser descrito (en términos relacionales) como el resultado de una proyección y/o una selección de la información contenida en la BD, de modo que resulten subconjuntos disjuntos de ella. Por tanto, una vez que la BD ha sido dividida en un conjunto de fragmentos, todos los datos aparecen en uno y sólo uno de estos fragmentos*
- *Fragmento físico: Subconjunto de un fragmento lógico que se almacena completo en un nodo. Cada uno de los fragmentos físicos de un fragmento lógico se localizan en uno o más nodos dependiendo del modelo de distribución de los datos.*

El esquema de fragmentación es el conjunto de las distintas relaciones que se han obtenido del ELG y las condiciones empleadas para esta división expresado en algebra relacional.

Existen distintas formas de fragmentar una relación [2]:

- Fragmentación vertical

- Fragmentación horizontal
- Fragmentación mixta

### **2.2.6.1 Razones para la fragmentación**

Los esquemas de fragmentación se diseñan teniendo en cuenta el uso que se va a hacer de los datos que se almacenan en cada una de las sedes, construyendo relaciones más pequeñas y más adaptadas a las operaciones de recuperación y actualización que utilizan las aplicaciones, tratando así de aligerar las comunicaciones entre los nodos debido al alto coste que éstas tienen [6].

Las cuatro razones para la fragmentación son las siguientes:

- Es útil ya que las aplicaciones de BD suelen funcionar con vistas, y por ello se pueden utilizar distintas relaciones en distintos nodos para formar la unidad distribuida.
- Se consigue una mayor eficiencia dado que los datos suelen estar almacenados cerca del nodo que más utiliza dichos datos.
- Permite aumentar el grado de concurrencia porque la fragmentación de las relaciones permite que una transacción pueda dividirse en subconsultas que operan sobre estos fragmentos.
- Aporta una mayor seguridad, dado que los datos no utilizados por un nodo local no se almacenan en él y por lo tanto no están al alcance para personas sin autorización.

Si las aplicaciones tienen requisitos conflictivos que impiden la descomposición de la relación en fragmentos mutuamente exclusivos, estas aplicaciones cuyas vistas se definen en más de un fragmento pueden sufrir una degradación del rendimiento.

El segundo problema está relacionado con el control semántico de los datos, especialmente para el control de integridad. Como resultado de la fragmentación, los atributos que participan en una dependencia pueden ser descompuestos en diferentes fragmentos que pueden ser asignados a diferentes sitios. En este caso, incluso la simple tarea de comprobar las dependencias daría lugar a la búsqueda de los datos en diferentes sitios.

### **2.2.6.2 Requisitos de información**

Uno de los aspectos del diseño de distribución es que contribuyen demasiados factores para conseguir un diseño óptimo [6]. La organización lógica de la base de datos, la ubicación de las aplicaciones, las características de las solicitudes de acceso a la base de datos, y las propiedades de los sistemas informáticos en cada sitio tienen influencia sobre las decisiones de distribución. Esto hace que sea muy complicado para formular un problema de distribución.

La información necesaria para el diseño de la distribución puede dividirse en cuatro categorías: información de la base de datos, la aplicación de la información, información de la red de comunicación, y sistema informático.

### 2.2.6.3 Alternativas de fragmentación

Esencialmente las relaciones son las tablas, así que la dificultad reside en encontrar diferentes formas de dividir una tabla en otras más pequeñas [6]. Es evidente que hay dos alternativas para esto: la fragmentación horizontal o la fragmentación vertical.

La fragmentación puede ser anidada. Si las fragmentaciones son de tipos diferentes, se obtiene la fragmentación mixta o híbrida. Aún cuando la fragmentación híbrida no es considerada como un tipo primitivo de las estrategias de fragmentación es bastante obvio que en muchas ocasiones de la vida real las particiones pueden ser híbridas.

### 2.2.6.4 Grado de fragmentación

La medida en que la base de datos debe ser fragmentada es una decisión importante que afecta el rendimiento de ejecución de las consultas [6]. El grado de fragmentación va de un extremo, es decir, no fragmentar nada, al otro extremo, a fragmentar el nivel de tuplas individuales (en el caso de la fragmentación horizontal) o al nivel de los atributos individuales (en el caso de la fragmentación vertical).

Lo que necesitamos es encontrar un nivel adecuado de fragmentación, que es un compromiso entre los dos extremos. Ese nivel sólo se puede definir con respecto a las aplicaciones que se ejecutan en la base de datos. La cuestión es ¿cómo? En general, las aplicaciones deben ser caracterizadas con respecto a una serie de parámetros. Los fragmentos individuales pueden ser identificados de acuerdo con los valores de estos parámetros.

### 2.2.6.5 Reglas de fragmentación

Hay que tener en cuenta que la fragmentación podría afectar al rendimiento del SGBDD, especialmente cuando se utilizan distintos fragmentos ubicados en distintos nodos para construir una vista. También el control de la semántica de los datos puede ser más complicado por la fragmentación.

Para asegurar que la BD no sufrirá cambios semánticos durante la fragmentación de los datos, se definen las siguientes tres normas que determinen la calidad de la fragmentación de una relación [4]:

**Compleitud:** Sea  $R$  una relación descompuesta en fragmentos  $F(R) = \{R_1, R_2, \dots, R_n\}$ . Entonces cada atributo de la relación deberá encontrarse en al menos uno de los fragmentos  $R_i$  asegurando así que no hay pérdida de información.

**Reconstrucción:** si una relación  $R$  se descomponen en fragmentos de la forma  $R_1, R_2, \dots, R_n$ , debería ser posible definir un operador relacional  $\nabla$  tal que  $R = \nabla R_i$  para todo  $R_i \in F_R$  (Fragmentos de  $R$ ).

El operador  $\nabla$  será diferente para cada forma de fragmentación; sin embargo es importante poder identificarlo. Esta regla asegura que las restricciones de integridad referencial están resguardadas.

**Disyunción:** si una relación R es fragmentada horizontalmente en  $R_1, R_2, \dots, R_n$ , y el dato  $d_i$  está en  $R_j$  y solo en  $R_j$ . Esta regla asegura que los fragmentos son disjuntos. Si se trata de una fragmentación vertical, sólo los atributos clave deben estar repetidos en todos los fragmentos, de modo que la propiedad disyunción se refiere únicamente a los restantes atributos.

### 2.2.6.6 Fragmentación vertical

La fragmentación vertical divide la relación verticalmente en columnas, así cada fragmento mantiene ciertos atributos de la relación original [2].

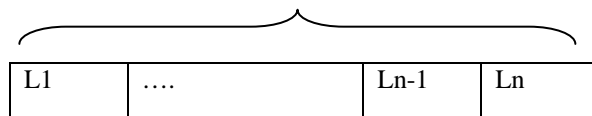
La fragmentación se realiza mediante la operación PROYECCIÓN:  $\prod_{L_i}(R)$  donde  $i=1 \dots n$  y  $R_i$  es el conjunto de fragmentos en que se divide la relación original.  $L_i$  es el criterio de fragmentación. En la Ilustración 2 se puede observar un esquema de la fragmentación vertical y en el Ejemplo 1 un caso de fragmentación vertical de una tabla.

Para que la fragmentación vertical sea correcta los subconjuntos de atributos  $L_i$  han de cumplir:

- La unión de todos los  $L_i$  contiene todos los atributos de la relación original.
- La intersección de todos los  $L_i$  consiste en la clave primaria de la relación original. Todos los  $L_i$  tienen en común la clave primaria para poder reconstruir R mediante JOIN.

$$L_1 \cap \dots \cap L_{n-1} \cap L_n = \text{Clave Primaria}$$

$$L_1 \cup \dots \cup L_{n-1} \cup L_n = L, \text{ todos los atributos de la relación}$$



**Ilustración 2. Esquema de una fragmentación vertical.**

#### Ejemplo 1 fragmentación vertical de una tabla

En la siguiente relación ALUMNOS (R)

IDALUMNO	NOMBRE	APELLIDOS	EDAD	CAMPUS	TITULACION	CURSO
101	María	Ruiz Olmos	20	Leganés	ITIG	2
102	Luís	Álvarez Delgado	21	Getafe	Derecho	2
103	Ana	Rosado Gil	19	Colmenarejo	Documentación	1
104	Benito	Sánchez Marín	25	Getafe	Economía	3

Suponiendo que en el campus de Getafe hubiera un nodo de la BD y otro nodo se ubicara en el campus de Leganés, y que por ejemplo el tratamiento y consulta de los datos personales se hiciera en el primer campus y el concerniente a los datos académicos se hiciera en el segundo, podría ser interesante realizar una fragmentación vertical en la que por un lado estuvieran:

- los datos personales con los campos “Nombre”, “Apellidos” y “Edad” ( $L_1$ ).
- y otra de datos académicos con los campos “Campus”, “Titulación” y “Cursos” ( $L_2$ ).



Así, se podría almacenar una subrelación (fragmento) en el campus de Getafe y otra en el campus de Leganés, quedando cada subrelación de la siguiente manera:

DATOS\_PERS: Fragmento vertical de datos personales ( $R_1$ ).

IDALUMNO	NOMBRE	APELLIDOS	EDAD
101	María	Ruiz Olmos	20
102	Luís	Álvarez Delgado	21
103	Ana	Rosado Gil	19
104	Benito	Sánchez Marín	25

DATOS\_ACAD: Fragmento vertical de datos académicos ( $R_2$ ).

IDALUMNO	CAMPUS	TITULACION	CURSO
101	Leganés	ITIG	2
102	Getafe	Derecho	2
103	Colmenarejo	Documentación	1
104	Getafe	Economía	3

### 2.2.6.7 Fragmentación horizontal

La fragmentación horizontal divide la relación en subconjuntos de tuplas, cada uno de ellos con un significado lógico [2].

La fragmentación se realiza mediante la operación SELECCIÓN:  $\sigma_{C_i}(R)$

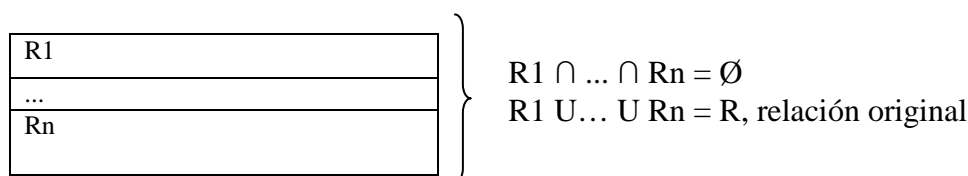
Existen dos tipos de fragmentación horizontal: primaria y derivada. En la Ilustración 3 se puede observar un esquema de la fragmentación horizontal y en Ejemplo 2 un caso de fragmentación horizontal de una tabla.

- Primaria, es una selección de la relación original. La fragmentación se realiza mediante la operación  $R_i = \sigma_{P_i}(R)$  donde  $P_i$  es un predicado sobre uno o más atributos de  $R$  y es la condición empleada para seleccionar el contenido de los fragmentos.
- Derivada, es una selección en función de predicados definidos sobre atributos de otras relaciones o fragmentos; esto se debe a que la relación  $R$  a fragmentar depende de la relación  $Q$ . Además,  $R$  hace referencia a  $Q$  mediante una clave ajena. La fragmentación se realiza mediante la operación  $R_i = R \bowtie Q_i$  donde  $Q_i$  corresponde al conjunto de fragmentos en los que se ha dividido la relación  $Q$ . La semicombinación ( $\bowtie$ ) se hace por el atributo que relacionan estas dos tablas.

Para que la fragmentación horizontal sea correcta se tiene que cumplir que:

- La unión de todos los  $R_i$  sea la relación original ( $R$ ).
- La intersección de todos los  $R_i$  sea vacía.

La relación original se recupera mediante UNION de los fragmentos.



**Ilustración 3. Esquema de una fragmentación horizontal.**

### Ejemplo 2 fragmentación horizontal de una tabla

En la siguiente relación ALUMNOS (R)

IDALUMNO	NOMBRE	CAMPUS
101	María Ruiz Olmos	Leganés
102	Luís Álvarez Delgado	Getafe
103	Ana Rosado Gil	Colmenarejo
104	Benito Sánchez Marín	Getafe

Podría ser interesante realizar una fragmentación horizontal por el campo “Campus” y almacenar cada fragmento en cada uno de los campus, suponiendo que en estos hubiera un nodo de la BD, quedando cada subrelación de la siguiente manera:

ALUMNOS\_LEG: Fragmento correspondiente a alumnos del campus de Leganés ( $R_1 = \sigma (\text{CAMPUS} = \text{Leganés})$ )

IDALUMNO	NOMBRE	CAMPUS
101	María Ruiz Olmos	Leganés

ALUMNOS\_GET: Fragmento correspondiente a alumnos del campus de Getafe ( $R_2 = \sigma (\text{CAMPUS} = \text{Getafe})$ )

IDALUMNO	NOMBRE	CAMPUS
102	Luís Álvarez Delgado	Getafe
104	Benito Sánchez Marín	Getafe

ALUMNOS\_COL: Fragmento correspondiente a alumnos del campus de Colmenarejo ( $R_3 = \sigma (\text{CAMPUS} = \text{Colmenarejo})$ )

IDALUMNO	NOMBRE	CAMPUS
103	Ana Rosado Gil	Colmenarejo

#### 2.2.6.8 Fragmentación mixta o híbrida

La fragmentación mixta es la mezcla de fragmentación horizontal y vertical. Hay dos tipos de fragmentación mixta [2]:

- Fragmentación VH  
Es una fragmentación vertical seguida de una fragmentación horizontal, sobre cada uno de los fragmentos verticales. Véase un esquema de este tipo de fragmentación el la Ilustración 4.

FV1Hn	FV2Hn		FVmHn	FVi = fragmentos verticales
...	...			
FV1H2	FV2H2			FViHj = fragmentos horizontales sobre fragmentos verticales
FV1H1	FV2H1			
FV1	FV2	...	FVm	

**Ilustración 4. Esquema de una fragmentación vertical-horizontal.**

### Ejemplo 3 fragmentación vertical-horizontal

En la siguiente relación ALUMNOS (R)

IDALUMNO	NOMBRE	APELLIDOS	EDAD	CAMPUS	TITULACION	CURSO
101	María	Ruiz Olmos	20	Leganés	ITIG	2
102	Luís	Álvarez Delgado	21	Getafe	Derecho	2
103	Ana	Rosado Gil	19	Colmenarejo	Documentación	1
104	Benito	Sánchez Marín	25	Getafe	Economía	3

Podría ser interesante realizar una fragmentación vertical en la que se partitionaran

- los datos personales con los campos “Nombre”, “Apellidos” y “Edad” (FV<sub>1</sub>)
- los datos académicos con los campos “Campus”, “Titulación” y “Cursos” (FV<sub>2</sub>)

Seguida de una fragmentación horizontal por el campo “Campus”.

De este modo se obtendrían los siguientes fragmentos....

Datos personales (FV<sub>1</sub>) del Campus Leganés (FV<sub>1</sub>H<sub>1</sub>), Campus Getafe (FV<sub>1</sub>H<sub>2</sub>) y Campus Colmenarejo (FV<sub>1</sub>H<sub>3</sub>)

y datos académicos (FV<sub>2</sub>) del Campus Leganés (FV<sub>2</sub>H<sub>1</sub>), Campus Getafe (FV<sub>2</sub>H<sub>2</sub>) y Campus Colmenarejo (FV<sub>2</sub>H<sub>3</sub>).

#### Fragmentación HV

Es una fragmentación horizontal seguida de una fragmentación vertical, sobre cada uno de los fragmentos horizontales. Véase un esquema de este tipo de fragmentación el la Ilustración 5.

FHn			FHnVm	FHi = fragmentos horizontales
...				
FH2				FHiVj = fragmentos verticales sobre fragmentos horizontales
FH1	FH1V1	FH1V2	FH1V3	

**Ilustración 5 Esquema de una fragmentación horizontal-vertical**

## **Ejemplo 4 fragmentación horizontal-vertical**

En la relación ALUMNOS podría ser interesante realizar una fragmentación vertical por el campo “Campus”

Campus de Leganés (FH<sub>1</sub>)

Campus de Getafe (FH<sub>2</sub>)

Campus de Colmenarejo (FH<sub>3</sub>)

Seguida de una fragmentación vertical (V1) en la que por un lado estuvieran

- los datos personales con los campos “Nombre”, “Apellidos” y “Edad” (campus de Leganés FH1V1, campus Getafe FH2V1 y campus Colmenarejo FH3V1)
- y otra (V2) de datos académicos con los campos “Campus”, “Titulación” y “Cursos” (campus de Leganés FH1V2, campus Getafe FH2V2 y campus Colmenarejo FH3V2).

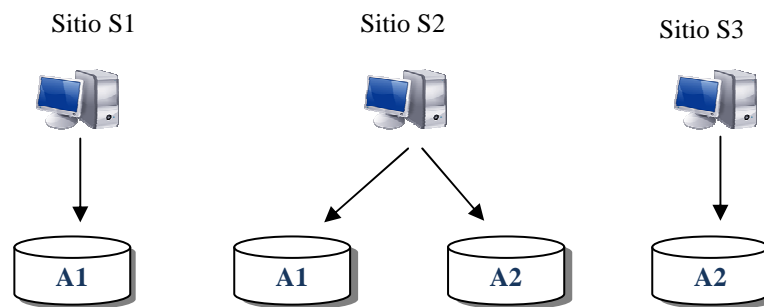
### **2.2.7 Asignación de fragmentos**

Cada fragmento (o cada copia de un fragmento) se debe asignar a un sitio determinado en el sistema distribuido. Este proceso se denomina distribución de los datos. La elección de la sede y el grado de replicación depende de los objetivos de rendimiento y disponibilidad para el sistema y de los tipos de frecuencias de transacciones introducidas en cada sitio. Por ejemplo, si se requiere una alta disponibilidad, y si las transacciones se pueden introducir en cualquier sitio y si la mayoría de ellas son de obtención de datos, entonces una BD completamente replicada será una buena opción. Sin embargo, si por lo regular ciertas transacciones que tienen acceso a partes específicas de la BD se introducen en un solo sitio, se podría asignar el conjunto de fragmentos correspondiente exclusivamente a ese sitio. Los datos que se utilizan en múltiples sitios se pueden replicar en esos sitios. Si se efectúan muchas actualizaciones, puede ser conveniente limitar la replicación. Encontrar una solución óptima, o siquiera buena, para el reparto de datos distribuidos es un problema de optimización muy complejo [1].

### **2.2.8 Alternativas de replicación**

La replicación de datos se refiere al almacenamiento de copias de datos en múltiples sitios conectados mediante una red de ordenadores. Pueden guardarse copias fragmentadas en varios sitios para satisfacer requerimientos de información específicos. Como la existencia de copias de fragmentos puede mejorar la disponibilidad de los datos y el tiempo de respuesta, estas copias reducen los costes de comunicación y de consultas totales [6].

Supóngase que la tabla A está dividida en dos fragmentos A1 y A2. Dentro de una base de datos distribuida replicada, es posible el escenario de la Ilustración 6 en el que el fragmento A1 se guarda en los sitios S1 y S2, mientras que el A2 se guarda en los sitios S2 y S3.



**Ilustración 6. Replicación de los datos**

Los datos replicados se someten a la regla de consistencia mutua. La regla de consistencia mutua requiere que todas las copias de fragmentos de datos sean idénticas. Por consiguiente, para mantener la consistencia de los datos entre las réplicas, el SGBDD debe garantizar que se realice una actualización de la base de datos en todos los sitios donde existan réplicas.

Suponiendo que la base de datos está fragmentada correctamente, hay que decidir la asignación de los fragmentos a diversos sitios en la red. Cuando se asignan los datos, o bien se pueden replicar o mantenerse una sola copia de los mismos. Las razones para la replicación son la disponibilidad, fiabilidad y la eficiencia de consultas de sólo lectura. Si hay múltiples copias de un elemento de datos, hay una buena probabilidad de que alguna copia esté accesible en algún lugar, incluso cuando se producen fallos en el sistema. Por otra parte, las consultas de sólo lectura que acceden a los mismos elementos de datos pueden ser ejecutados en paralelo desde la salida de copias en múltiples sitios.

Por otra parte, la ejecución de las transacciones de actualización causa problemas ya que el sistema tiene que garantizar que todas las copias de los datos se actualizan correctamente. De ahí que la decisión con respecto a la replicación dependa de la relación existente entre consultas de sólo lectura y consultas de actualización. Esta decisión afecta a casi todos los algoritmos de SGBDD y las funciones de control.

Una base de datos no replicada (comúnmente se llama base de datos particionada) contiene fragmentos que se asignan a los diferentes sitios, y sólo hay una copia de cada fragmento en la red. En el caso de realizarse replicación, la base de datos puede ubicarse en su totalidad en cada lugar (base de datos totalmente replicada), o los fragmentos pueden distribuirse en los nodos de tal manera que las copias de un fragmento puedan residir en varios emplazamientos (base de datos parcialmente replicada). En este último caso el número de copias de un fragmento puede suponer un aumento de coste para el algoritmo de asignación.

En la Tabla 1 se comparan estas alternativas con respecto a varias funciones de los SGBD distribuidos.

	<b>Totalmente replicadas</b>	<b>Parcialmente replicada</b>	<b>Particionada</b>
<b>Proceso de consultas</b>	Fácil	Misma dificultad	
<b>Gestión de directorios</b>	Fácil o no existe	Misma dificultad	
<b>Control de concurrencia</b>	Moderada	Difícil	Fácil
<b>Fiabilidad</b>	Muy alta	Alta	Baja
<b>Realidad</b>	Posible aplicación	Realista	Posible aplicación

**Tabla 1 Comparación de las alternativas de replicación**

Aunque la replicación tiene algunos beneficios, también exige más complejidad de procesamiento del SGBDD, porque cada copia de datos debe ser mantenida por el sistema. Para ilustrar la complejidad de un SGBDD, se deben considerar los procesos que el SGBDD debe realizar a la hora de gestionar la base de datos:

- Si la base de datos está fragmentada, el SGBDD debe decidir qué copia acceder.
- Una operación de lectura selecciona la copia más cercana para satisfacer la transacción. Una operación de escritura requiere que todas las copias se seleccionen y actualicen para satisfacer la regla de consistencia mutua.
- El procesador de transacciones envía una solicitud de datos a cada procesador de datos para su ejecución.
- El procesador de datos recibe y ejecuta cada solicitud y envía los datos de vuelta al procesador de transacciones.
- El procesador de transacciones construye las respuestas del procesador de datos.

El problema se complica más cuando se consideran factores adicionales tales como la topología de red y los protocolos de comunicación.

## **2.3 Control semántico de los datos**

La integridad semántica de los datos se refiere a las características del SGBD y los procesos que se pueden utilizar para asegurar la exactitud y la viabilidad del contenido de los datos de una BD. La integridad semántica de los datos se refiere a la consistencia de los datos en sí.

En general, se utilizan las funciones del SGBD para apoyar la integridad de los datos que por lo general ofrece la mejor solución. Es más fácil descubrir y mantener una regla de integridad semántica utilizando las funciones del SGBD que revisando el código del programa. Hay que tener en cuenta que el código puede estar en diferentes lenguajes en un entorno heterogéneo. Además, habrá que distinguir los datos obsoletos de los activos [10].

## 2.4 Procesamiento de consultas

### 2.4.1 Definiciones

#### **Esquema conceptual.**

*Describe la estructura de toda la base de datos para una comunidad de usuarios. El esquema conceptual oculta los detalles de las estructuras físicas de almacenamiento y se concentra en describir entidades, tipos de datos, vínculos, operaciones de los usuarios y restricciones [1].*

#### **Esquema Conceptual Local (ECL).**

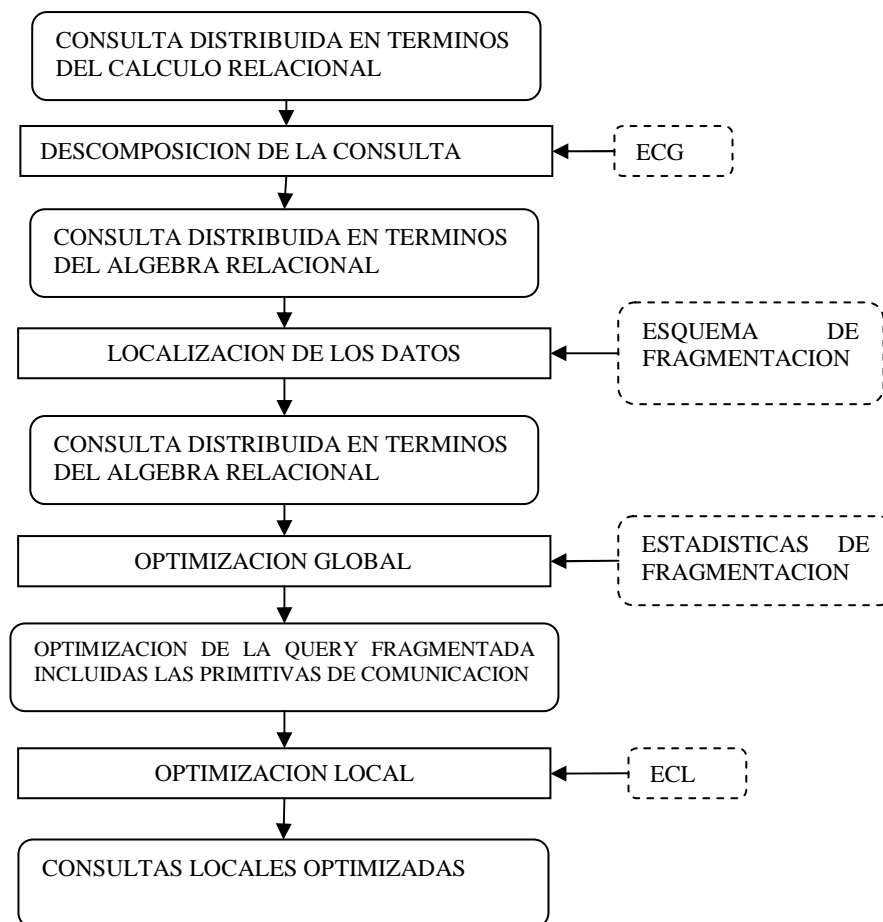
Son los esquemas conceptuales correspondientes a cada vista de usuario.

#### **Esquema Conceptual Global (ECG).**

*En el caso de SGBDDs lógicamente integrados, el ECG define la visión global de la BD completa. El ECG es la unión de los Esquemas Conceptuales Locales (ECL) [4].*

El procesamiento de consultas se puede descomponer en un número de subconsultas a los que corresponden cada uno de los niveles de un procesador de consultas genérico. En la Ilustración 7 se puede ver esta arquitectura. La entrada es una consulta sobre datos distribuidos que se expresa en términos del cálculo relacional. Por tanto, se refiere a relaciones globales, distribuidas en los nodos de forma transparente. Es necesario traducir la consulta distribuida en un número de operaciones locales, en cada BD local, que se optimizarán separadamente. Para ello se establecen cuatro niveles que se reparten de la siguiente manera:

- La descomposición de la consulta, la localización de los datos y la optimización global de la consulta se ejecutan en el nodo central y emplean información global.
- La optimización local de la consulta, tiene únicamente ámbito local, en cada nodo.



**Ilustración 7. Procesamiento de una Consulta Distribuida**

### 2.4.2 Descomposición de consultas globales.

Este nivel tiene como objetivo simplificar la consulta y reescribirla en términos de álgebra relacional [4]. La información necesaria para esta transformación se encuentra en la definición del ECG que describe las relaciones globales. La información sobre la distribución de los datos en el ECG no se usa hasta el siguiente nivel, de manera que las técnicas utilizadas son las mismas que en sistemas centralizados.

La descomposición puede verse como una secuencia de pasos sucesivos:

- Reescritura de la consulta en la forma más conveniente para el tratamiento posterior. Normalmente implica la manipulación de los cuantificadores conforme a la prioridad establecida para cada operador relacional.
- La consulta es entonces analizada semánticamente para detectar errores y en tal caso rechazarla lo antes posible. Las técnicas actuales para detectar errores sólo alcanzan a un subconjunto de las expresiones del cálculo relacional y típicamente se basan en algún grafo que captura la semántica de la consulta.



- La consulta (correcta) es simplificada por eliminación de redundancia, lo cual es muy probable que suceda debido a las anteriores transformaciones.
- Finalmente, la consulta se expresa en términos del álgebra relacional. La calidad de la expresión resultante se mide en base al rendimiento esperado. Este es a menudo un proceso de refinamientos sucesivos.

### 2.4.3 Localización de los datos.

La entrada a este nivel es una consulta expresada en términos del álgebra relacional y sobre relaciones distribuidas [4]. Esta expresión se transforma en una consulta fragmentada equivalente, expresada sobre fragmentos almacenados en diferentes nodos que a su vez puede ser nuevamente simplificada por la aplicación de las reglas del álgebra relacional. Lo principal es, por tanto, localizar los datos para la consulta a partir de la información relativa a la distribución de los mismos. Este nivel determina que fragmentos están implicados en la resolución de la consulta y la convierte así en una consulta fragmentada.

La fragmentación de la consulta tiene lugar en dos etapas que resultan de aplicar las reglas de fragmentación primero y entonces derivar un programa de localización que consiste en operaciones del álgebra relacional sobre los fragmentos. Estas etapas son:

- La consulta distribuida se traduce en una consulta fragmentada por sustitución de cada relación distribuida por el correspondiente fragmento. Este proceso corresponde a un programa denominado programa de materialización y que se ejecuta después del programa de localización.
- La consulta fragmentada es simplificada y reestructurada para producir otra consulta mejorada. Las reglas de simplificación pueden ser las mismas que en el nivel anterior y dan lugar a un procedimiento de refinamiento progresivo.

#### 2.4.3.1 Optimización de consultas globales.

La optimización global de la consulta genera un plan de ejecución óptimo para la entrada a este nivel: la consulta fragmentada expresada sobre fragmentos. La estrategia de ejecución de una consulta distribuida puede describirse en base a operaciones del álgebra relacional más primitivas de comunicaciones (*send/receive*) para transferencia de datos entre los nodos. Las modificaciones consisten en reordenar las operaciones teniendo en cuenta las operaciones de comunicación entre nodos como el factor más determinante, además de los habituales (espacio en disco, entrada/salida, consumo de CPU, etc.) y en la selección de los algoritmos más convenientes. Los niveles previos han efectuado ya optimizaciones que son independientes de las características de los fragmentos, tales como su cardinalidad, y de las primitivas de comunicaciones.

Para seleccionar el orden más adecuado es necesario predecir los costes de ejecución de las ordenaciones alternativas, lo cual requiere estadísticas de los fragmentos y heurísticas que permitan estimar las cardinalidades de los resultados. Por tanto, el modelo de costes es central en la optimización global en tanto que provee la necesaria

abstracción de la ejecución por el SGBDD, en términos de métodos de acceso fundamentalmente, y de la BDs en términos de esquemas y estadísticas. A menudo el modelo conlleva importantes restricciones que limitan la efectividad de la optimización, en aras normalmente de la manejabilidad del conjunto. Un aspecto particularmente importante es el de ordenación del join, pues las permutaciones de los joins pueden mejorar drásticamente el rendimiento. Por esto mismo, es habitual emplear el semijoin para reducir el tamaño de los operandos del join y, consecuentemente, el coste de la comunicación entre nodos.

Aunque los lenguajes de consulta son progresivamente más potentes, la optimización global de consultas se focaliza sobre el llamado subconjunto SPJ esto es, consultas SELECT-PROJECT-JOIN con predicados conjuntivos. Esta es una clase importante de consultas para las que existen buenas opciones de optimización. No ocurre esto mismo con las consultas que incluyen disyunciones, uniones, agregados o algún tipo de ordenación, debido sobre todo a que las optimizaciones tradicionales no pueden ser fácilmente extendidas al modelo distribuido. Por ejemplo, la función AVERAGE sobre una relación no produce idéntico resultado que sobre los correspondientes fragmentos. Una posible solución consistiría en centralizar los operandos previamente al procesamiento de la consulta pero ello elimina el posible paralelismo intraconsulta.

La optimización global de una consulta normalmente tiene lugar previamente a la ejecución de la misma. En este sentido se dice que es estática pues puede efectuarse por el compilador y puede además explotar el conocimiento del esquema y de la localización de los datos. Obviamente esto resulta afectado por cualquier cambio o reorganización de la base de datos. El problema ahora consiste en determinar el intervalo de recompilación de las consultas.

#### **2.4.3.2 Optimización de consultas locales.**

A este nivel se emplea el ECL en cada nodo y los algoritmos son los típicos de sistemas relacionales centralizados. La optimización es, a diferencia de las optimizaciones globales, normalmente dinámica, en tiempo de ejecución de la consulta [4].

### **2.5 Control de concurrencia en BDD**

El control de concurrencia trata con los problemas de aislamiento y consistencia del procesamiento de transacciones. El control de concurrencia distribuido de un SGBDD asegura que la consistencia de la BD se mantiene en un ambiente distribuido multiusuario. Si las transacciones son internamente consistentes, la manera más simple de lograr este objetivo es ejecutar cada transacción individualmente, una después de otra. Sin embargo, esto puede afectar en gran medida al desempeño de un SGBDD dado que el nivel de concurrencia se reduce al mínimo. El nivel de concurrencia, el número de transacciones activas es probablemente el parámetro más importante en sistemas distribuidos. Por lo tanto, los mecanismos de control de concurrencia buscan encontrar un balance entre el mantenimiento de la consistencia de la BD y el mantenimiento de un alto nivel de concurrencia.

Si no se hace un adecuado control de concurrencia, se pueden presentar dos problemas:

- Se pueden perder actualizaciones provocando que los efectos de algunas transacciones no se reflejen en la base de datos.
- Se pueden presentar recuperaciones de información inconsistentes.

En este apartado se hace la suposición de que el sistema distribuido es completamente confiable y no experimenta fallos.

### Teoría de la seriabilidad

Un *programa* se define sobre un conjunto de transacciones  $T = \{T_1, T_2, \dots, T_n\}$  y especifica un orden entrelazado de la ejecución de las operaciones de las transacciones. El programa puede ser especificado como un orden parcial sobre  $T$ .

**Ejemplo 5 Teoría de la seriabilidad:** Considere las siguientes tres transacciones:

$T_1$ : Read( $x$ )	$T_2$ : Write( $x$ )	$T_3$ : Read( $x$ )
Write( $x$ )	Write( $y$ )	Read( $y$ )
Commit	Read( $z$ )	Read( $z$ )
	Commit	Commit

Un programa con las acciones de las tres transacciones anteriores puede ser:

$$H_1 = \{ W_2(x), R_1(x), R_3(x), W_1(x), C_1, W_2(y), R_3(y), R_2(z), C_2, R_3(z), C_3 \}$$

Dos operaciones  $O_{ij}(x)$  y  $O_{kl}(x)$  ( $i$  y  $k$  no necesariamente distintos) que acceden al mismo dato de la BD  $x$  se dice que están en conflicto si al menos una de ellas es de escritura. De esta manera, las operaciones de lectura no tienen conflictos consigo mismas. Existen dos tipos de conflictos:

- de lectura-escritura (o escritura-lectura)
- de escritura-escritura.

Por tanto el orden de ejecución entre dos operaciones en conflicto es importante.

Un programa completo define el orden de ejecución de todas las operaciones en su dominio.

Si en un programa  $S$ , las operaciones de varias transacciones no están entrelazadas, entonces se dice que el programa es *serial*. Si cada transacción es consistente (obedece las reglas de integridad), entonces se garantiza que la BD será consistente al final del programa serial. La historia asociada a este tipo de programa se le conoce como *serial*.

Ejemplo: La siguiente es una historia serial para el Ejemplo 5:

$$H_S = \{ W_2(x), W_2(y), R_2(z), C_2, R_1(x), W_1(x), C_1, R_3(x), R_3(y), R_3(z), C_3 \}$$

Las transacciones se ejecutan de manera concurrente, pero el efecto neto de la historia resultante sobre la BD es *equivalente* a alguna *historia serial*. Basada en la relación de

precedencia introducida por el orden parcial, es posible discutir la equivalencia de diferentes programas con respecto a sus efectos sobre la base de datos.

Dos programas,  $S_1$  y  $S_2$ , definidos sobre el mismo conjunto de transacciones  $T$ , se dice que son *equivalentes* si para cada par de operaciones en conflicto  $O_{ij}$  y  $O_{kl}$  ( $i \neq k$ ), cada vez que  $O_{ij} <_1 O_{kl}$ , entonces,  $O_{ij} <_2 O_{kl}$ . A esta relación se le conoce como *equivalencia de conflictos* puesto que define la equivalencia de dos programas en término del orden de ejecución relativo de las operaciones en conflicto en ellas.

Un programa  $S$  se dice que es *serializable*, si y solamente si, es equivalente por conflictos a un programa serial.

Ejemplo: Los siguientes programas no son equivalentes por conflicto:

$$H_S = \{ W_2(x), W_2(y), R_2(z), C_2, R_1(x), W_1(x), C_1, R_3(x), R_3(y), R_3(z), C_3 \}$$

$$H_1 = \{ W_2(x), R_1(x), R_3(x), W_1(x), C_1, W_2(y), R_3(y), R_2(z), C_2, R_3(z), C_3 \}$$

Los siguientes programas son equivalentes por conflictos; por lo tanto  $H_2$  es serializable:

$$H_S = \{ W_2(x), W_2(y), R_2(z), C_2, R_1(x), W_1(x), C_1, R_3(x), R_3(y), R_3(z), C_3 \}$$

$$H_2 = \{ W_2(x), R_1(x), W_1(x), C_1, R_3(x), W_2(y), R_3(y), R_2(z), C_2, R_3(z), C_3 \}$$

La función primaria de un controlador de concurrencia es generar un programa serializable para la ejecución de todas las transacciones. El problema es, entonces, desarrollar algoritmos que garanticen que únicamente se generan programas serializables.

## 2.6 Control de recuperación

Como ocurre con el control de concurrencia, en el control de recuperación surgen numerosos problemas que no surgen en SGBD centralizados, como [1]:

- Manejar múltiples copias de los elementos de información: el método de recuperación debe cuidar que una copia sea consistente con todas las demás si el sitio en el que la copia estaba almacenada falla y se recupera posteriormente.
- Fallo de sitios individuales: El SGBD debe continuar operando con sus sitios activos, si es posible, cuando fallen uno o más sitios individuales. Cuando un sitio se recupere, su BD local se deberá poner al día con los demás sitios antes de que se reincorpore al sistema.
- Fallo de enlaces de comunicaciones: El sistema debe ser capaz manejar el fallo de uno o más de los enlaces de comunicaciones entre los sitios. Un caso extremo de este problema es que puede haber partición de la red. Esto divide los sitios en dos o más particiones dentro de las cuales los sitios pueden comunicarse entre sí, pero no con sitios de otras particiones.
- Confirmación distribuida: Puede haber problemas para confirmar una transacción que está teniendo acceso a BD almacenadas en múltiples sitios si alguno de estos fallan durante el proceso de confirmación. A menudo se utiliza el protocolo de confirmación en dos fases (*two phase commit*) para resolver este problema.

Las técnicas de control de recuperación deben resolver estos problemas. El proceso de recuperación de las BDD es bastante complicado. En ciertos casos incluso es bastante difícil determinar si un sitio está inactivo sin intercambiar un gran número de mensajes con otros sitios. Por ejemplo, supóngase que el sitio X envía un mensaje al sitio Y y espera una respuesta de Y pero no la recibe. Hay varias explicaciones posibles:

- El mensaje no llegó a Y debido a un fallo de comunicación.
- El sitio Y está inactivo y no pudo responder.
- El sitio Y está activo y envió una respuesta, pero ésta no llegó.

Sin información adicional y sin el envío de mensajes adicionales, es difícil determinar qué sucedió realmente.

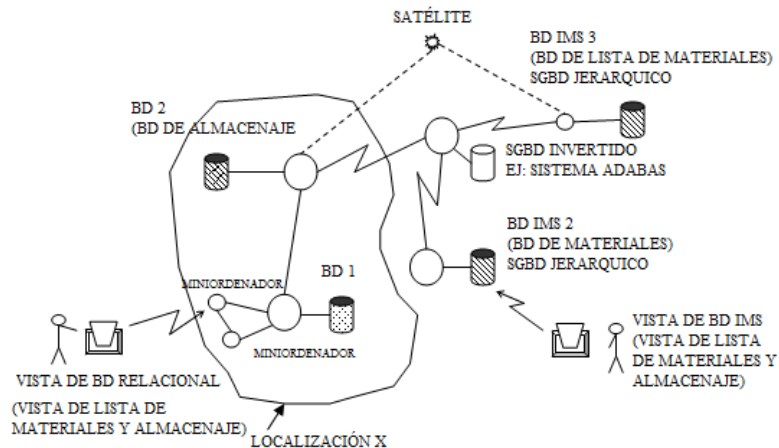
Otro problema con la recuperación distribuida es la confirmación distribuida. Cuando una transacción está actualizando datos en varios sitios, no puede confirmarse hasta estar segura de que el efecto de la transacción en todos los sitios no puede perderse. Esto significa que cada sitio debe haber asentado primero permanentemente los efectos locales de la transacción en la bitácora local en el disco del sitio. A menudo, se usa el protocolo de confirmación en dos fases, para garantizar la corrección de la confirmación distribuida.

## ***2.7 Bases de Datos Distribuidas Heterogéneas***

La proliferación de diferentes SGBD y los avances en la creación de redes de informática y las comunicaciones han llevado a aumentar los entornos de SGBD distribuidos heterogéneos de red. Los principales problemas y desafíos de la heterogeneidad son:

- diferentes modelos de base de datos,
- diferentes SGBD en términos sintácticos y semánticos,
- diferentes tipos de controles (de control de actualizaciones, recuperación, etc.),
- etc.

La utilización generalizada de diferentes SGBD ha proliferado en los últimos años. Como resultado, ha surgido el escenario de SGBDD heterogéneo. Un ejemplo se muestra en la Ilustración 8.



**Ilustración 8 Ejemplo de un escenario de un SGBD Heterogéneo**

El entorno de BBDD heterogéneas ha surgido en muchas organizaciones debido a [7]:

- a) la proliferación de BBDD
- b) la proliferación de diferentes SGBD
- c) la proliferación de una variedad de minicomputadoras y computadoras personales
- d) la aparición de las redes de comunicación, junto con el hardware y el software heterogéneo.
- e) los avances en las comunicaciones de datos
- f) bases de datos distribuidas
- g) la falta en general (no sólo local) de la planificación y el control de la base de datos.

## 2.7.1 Tipos de heterogeneidades

Los tipos de heterogeneidades en los sistemas de bases de datos pueden ser clasificadas en [8]:

- las debidas a las diferencias en el SGBD
- las debidas a las diferencias en la semántica de los datos.

### 2.7.1.1 Heterogeneidades debido a diferencias en los SGBD

La heterogeneidad debida a la utilización de diversos SGBDs es común en organizaciones que crecen sin una planificación en cuanto a sus sistemas de información. Dichos sistemas evolucionan paulatinamente en diferentes SGBDs o diferentes modelos de conceptualización tales como: jerárquico, de red, relacional u orientado a objetos. Diferentes departamentos dentro de la empresa pueden tener requerimientos diferentes y pueden seleccionar diferentes SGBDs. También los SGBDs adquiridos a lo largo de un período de tiempo pueden ser diferentes debido a los cambios en la tecnología. Cada SGBD tiene un modelo de datos subyacente utilizado para definir estructuras de datos y las restricciones. Tanto los aspectos de representación (estructura y restricciones) como de lenguaje pueden dar lugar a la heterogeneidad.

#### Diferencias en la estructura:

Diferentes modelos de datos proporcionan diferentes primitivas estructurales. Pueden darse cuatro tipos de conflictos: de tipo, de dependencia, de clave o de comportamiento. Los conflictos de tipo se producen cuando el mismo objeto es representado por un atributo en un esquema y por una entidad en otro. Los problemas de dependencia se

producen cuando los distintos modos de relación (por ejemplo, una relación uno a uno frente a una de muchos a muchos) se usan para representar lo mismo en diferentes esquemas. Los conflictos de clave se producen cuando existen varias claves candidatas disponibles y se seleccionan claves primarias diferentes en distintos esquemas. Los conflictos de comportamiento están implícitos en los mecanismos de modelado. Por ejemplo, borrar el último elemento de una BD puede provocar la eliminación de la entidad que contiene (es decir, la eliminación del último empleado puede causar la disolución del departamento). Si el contenido de la información no es el mismo, puede ser muy difícil solventar las diferencias [6], [8].

#### **Diferencias en las restricciones:**

Dos modelos de datos pueden soportar diferentes restricciones. Esto quiere decir que puede ser que en un modelo de datos existan  $x$  tipos de restricciones y en otro modelo  $y$ , que posiblemente sean diferentes. Los disparadores (o algún otro mecanismo) deben ser utilizados en los sistemas relacionales para captar la semántica de estos.

#### **Diferencias en los lenguajes de consulta:**

Como por ejemplo cuando se utilizan diferentes lenguajes de consulta para manipular los datos representados en modelos de datos diferentes. Incluso cuando dos SGBDs se basan en el mismo modelo de datos, las diferencias en los lenguajes de consulta (por ejemplo, Quel<sup>1</sup> y SQL) o las diferentes versiones de SQL soportado por dos SGBDs relacionales pueden contribuir a la heterogeneidad.

### **2.7.1.2 Heterogeneidades debido a diferencias semánticas**

La heterogeneidad semántica es un término bastante cargado, sin una definición clara. Básicamente se refiere a las diferencias entre las BBDD que se relacionan con el significado, la interpretación y el uso previsto de los datos. Sin duda, los aspectos más importantes de la heterogeneidad semántica se revelan como conflictos de nombres. El problema fundamental de nombres es el de los sinónimos y los homónimos. Dos entidades idénticas que tienen diferentes nombres son sinónimos, y dos entidades diferentes que tienen nombres idénticos son homónimas.

La detección de la heterogeneidad semántica es un problema difícil. Normalmente, los esquemas de los SGBD no proporcionan la semántica suficiente para interpretar los datos de forma coherente. La heterogeneidad debida a diferencias en los modelos de datos también contribuye a la dificultad en la identificación y resolución de la heterogeneidad semántica. También es difícil descomponer la heterogeneidad debida a diferencias en el SGBD de los resultantes de la heterogeneidad semántica [6] [7].

### **2.7.2 Ventajas de las BBDD en sistemas heterogéneos**

La llegada de los sistemas heterogéneos ha traído tanto beneficios como problemas. Las ventajas son por supuesto muchas.

1. El tradicional procesamiento de los datos está siendo rápidamente reemplazado por la gestión de modernas BBDD.

---

<sup>1</sup> Este lenguaje está basado en el cálculo relacional orientado a tuplas. Cuando nos queremos referir al atributo A de la tupla t, lo hacemos escribiendo t.A. [7]

2. Nuevas capacidades tales como la actualización en línea, consultas ad-hoc, y la integridad de los datos son fácilmente disponibles en SGBD modernos.
3. Para todas las aplicaciones posibles de uso intensivo y grandes volúmenes de datos, existe un SGBD para la aplicación.
4. Para una organización con diversas aplicaciones, un gran número de SGBD monomodelo y monolenguaje pueden ser utilizados para cada aplicación distinta.

### 2.7.3 Problemas de BBDD en sistemas heterogéneos

Sin embargo, hay problemas que pueden afectar a la eficacia y la eficiencia de la utilización de SGBD modernos en una gran organización, como pueden ser:

- **Difícil intercambio, control y utilización global de datos.** Si una organización tiene diferentes bases datos, con diferentes SGBD y diferentes modelos de datos, a la hora de acceder a la información es posible que los usuarios no sean expertos en diferentes modelos y SGBD. De hecho lo general es encontrarse con profesionales en un modelo y en un lenguaje concreto.

- **Alto Coste de mantenimiento y soporte.** Los SGBD heterogéneos en una gran organización tienden a ser muchos y sus BD de gran volumen, cada SGBD monomodelo y monolenguaje requiere el apoyo de un conjunto exclusivo de los ordenadores, un sistema separado de las unidades de disco, y un equipo dedicado de profesionales. Ya sea centralizado o distribuido, el conjunto de equipos, el sistema de discos, y el equipo de profesionales suponen una inversión importante y pesada para un SGBD monomodelo y monolenguaje.

Como el número de estos SGBD monomodelo y monolenguaje aumentan, los gastos se multiplican. Además, los costos de la actualización de equipos, SGBD mejora del rendimiento, y la capacidad de crecimiento de la base de datos puede tener que ser multiplicada también. Desde el punto de vista de mantenimiento y actualización, siempre es más caro y requiere mayor esfuerzo mantener o actualizar los sistemas múltiples que actualizar un sistema único.

- **Las similitudes y la multiplicidad de SGBD heterogéneos [9].** En resumen, los SGBD heterogéneos nos ofrecen soluciones informatizadas para aplicaciones de datos voluminosos e intensivos. En consecuencia, el procesamiento de los datos tradicionales y las nuevas aplicaciones de datos intensivos y voluminosos han llevado a la proliferación y la introducción de muchos SGBD monomodelo y monolingües. Si bien esta proliferación e introducción son gratificantes y probablemente continuarán, los SGBD monomodelos y monolingües, sin embargo presentan nuevos problemas e introducen otras cuestiones. Estos problemas y cuestiones son intrínsecas a la similitud de SGBD monomodelo y monolingües. Son identificados como:

- El primer problema, en el intercambio global de los datos, el control y la utilización es lo más fundamental, ya que impide que el usuario de un SGBD monomodelo y monolingüe utilice las BBDD de otros SGBD monomodelo y monolingües. En consecuencia, la carga se encuentra en la parte de los usuarios que deben aprender los otros modelos y lenguajes para tener acceso a otras BBDD y para la utilización de otros SGBD monomodelo y monolingües.



- El segundo problema es el coste. A medida que la popularidad y la funcionalidad de los SGBD son cada vez más evidentes, el uso de SGBD en una determinada organización se hará más frecuente. En consecuencia, habrá muchos SGBDs heterogéneos en una organización. Como SGBDs separados, monomodelos, y monolingües, que requieren apoyos por separado y hardware más potente que soporte este tipo de sistemas, lo que se traduce en mas costes.

#### **2.7.4 Integración de las BBDD.**

La integración de BBDD implica un proceso por el cual la información de las BBDD fragmentadas puede ser conceptualmente integradas de forma coherente en una única BD. En otras palabras, es el proceso de diseño del ECG. La integración de BD puede darse en dos pasos:

- Esquema de traducción
- Esquema de integración

##### **2.7.4.1 Esquema de traducción.**

El esquema de traducción es la tarea de asignación de un esquema a otro. Los esquemas de los componentes de BD se traducen en una representación canónica intermedia común (InS1, InS2,...,InSn). El uso de una representación canónica facilita los procesos de traducción por reducción del número de traductores que necesitan ser escritos. Esto requiere la especificación de un modelo de datos de destino para la definición del ECG. La elección del modelo canónico es importante. En principio, debe ser uno suficientemente expresivo para incorporar los conceptos disponibles en todas las BBDD que luego serán integradas. Un modelo de objetos generalmente se considera como el modelo canónico más adecuado.

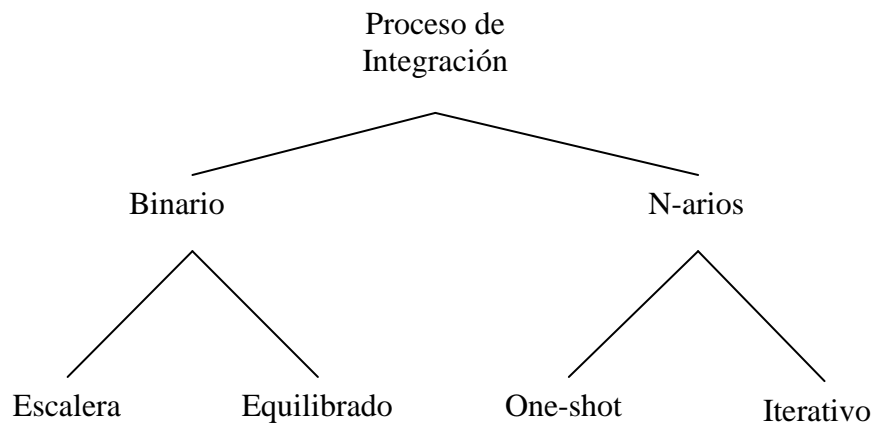
El paso de traducción es necesario sólo si los componentes de BBDD son heterogéneos y cada esquema local puede definirse mediante un modelo de datos diferente y aun así, puede no ser necesario en una BD heterogénea, si puede llevarse a cabo durante la etapa de integración. La combinación de la traducción y las medidas de integración, las establece el integrador con toda la información sobre la BD global en su totalidad a la vez. Obviamente, el integrador puede establecer compromisos entre los diferentes esquemas locales para determinar la representación a la que debe darse prioridad cuando surgen conflictos. Esto requiere que el integrador tenga conocimiento de todas las distintas ventajas y desventajas que deben hacerse entre varios esquemas diferentes y su semántica, que puede ser diferente.

##### **2.7.4.2 Esquema de integración.**

El esquema de integración sigue al proceso de traducción y genera el ECG mediante la integración de los esquemas intermedios. El esquema de integración es el proceso de identificación de los componentes de una BD que están relacionados entre sí, de la selección de la mejor representación para el ECG y, por último, de la integración de los componentes de cada esquema intermedio. Dos componentes pueden estar

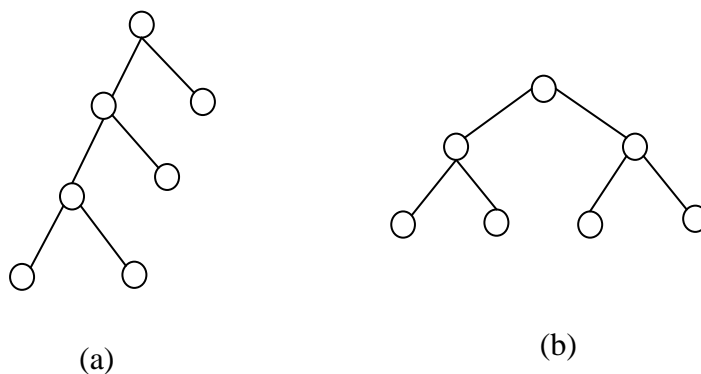
relacionados como equivalentes, uno está contenido en otro, o como disjunto.

Las metodologías de integración pueden ser clasificadas como mecanismos binarios o n-arios (Ilustración 9).



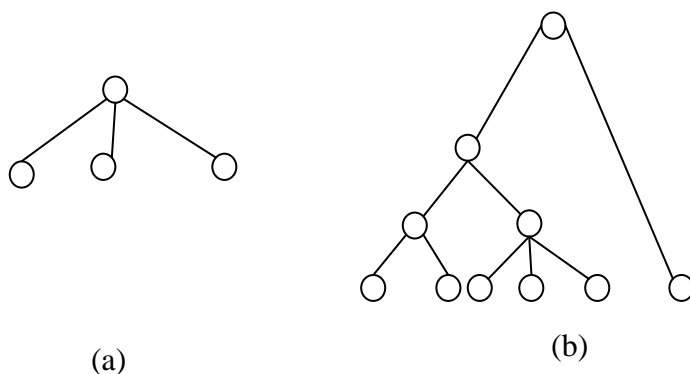
**Ilustración 9 Taxónomías de metodologías de la integración**

- Las metodologías de integración binarias implican la manipulación de dos esquemas a la vez. Ilustración 10. Se puede producir creando esquemas intermedios para la integración con los esquemas posteriores o de una manera puramente binaria.



**Ilustración 10 Métodos de integración binarios**

- Los mecanismos n-arios de integración integran más de dos esquemas en cada iteración. Un paso de integración (Ilustración 11) se produce cuando todos los esquemas se integran a la vez, produciendo el ECG después de una iteración. Los beneficios de este enfoque son la disponibilidad de información completa sobre todas las BD en tiempo de integración. Las dificultades con este enfoque incluyen el aumento de la complejidad y la dificultad de automatización.



**Ilustración 11 Métodos de Integración n-aria**

El mecanismo iterativo n-ario de integración (Véase la Ilustración 11) ofrece más flexibilidad (por lo general, está disponible más información) y es más general (el número de esquemas pueden variar en función de las preferencias de los integradores). Los modelos binarios son un caso especial de iterativos n-arios. Ellos disminuyen la complejidad de la integración potencial y conducen a las técnicas de automatización, ya que el número de esquemas que se examinarán en cada paso es más manejable. La integración por un proceso de n-arios permite al integrador realizar las operaciones en más de dos esquemas. Por razones prácticas, la mayoría de los sistemas utilizan la metodología de binarios.

La integración de esquemas implica dos tareas:

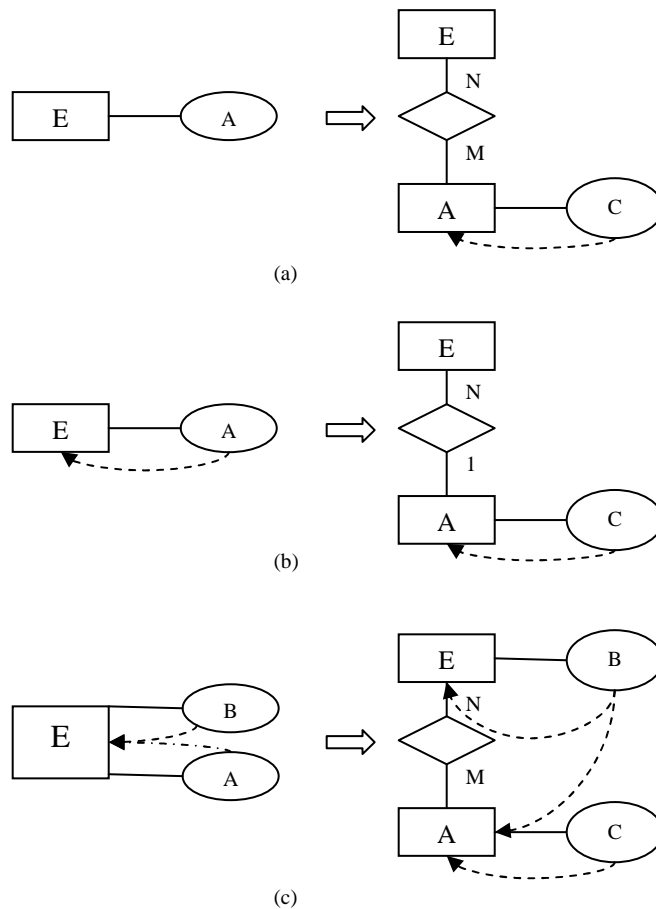
- la homogeneización
- la integración

### **Homogeneización**

Durante esta fase, se resuelven los problemas de la heterogeneidad semántica y de la heterogeneidad estructural.

Respecto a la heterogeneidad semántica, hay una serie de métodos alternativos para hacer frente a los conflictos de nombres. Uno de estos métodos es para resolver los homónimos anteponiendo los términos del esquema o nombre del modelo. No es posible resolver los sinónimos de una manera similar, simple.

Respecto a la heterogeneidad estructural, la transformación de las entidades y atributos / relaciones entre unos y otros es una manera de manejar los conflictos estructurales. La Ilustración 12 muestra los posibles escenarios de transformación atómica. Las líneas punteadas indican que un atributo dado es un identificador (clave) de la entidad asociada.



**Ilustración 12 Alternativas de Conformación Atómica**

La determinación de sinónimos y homónimos, así como la identificación de los conflictos estructurales, requiere la especificación de la relación entre los esquemas intermedios.

La determinación del tipo de relación es esencial en el diseño del esquema conceptual global. Por desgracia, la homogeneización requiere una cantidad significativa de la intervención humana, ya que se requiere el conocimiento semántico acerca de todos los esquemas intermedios [6].

### **Integración**

La integración supone la fusión de los esquemas intermedios y su reestructuración. Todos los esquemas se fusionan en un esquema de BD único y reestructurado para crear el "mejor" esquema integrado. La fusión requiere que la información contenida en los esquemas de participación se mantenga en el esquema integrado.

Tres dimensiones de la fusión y reestructuración pueden ser definidos: la integridad, minimalidad, y comprensibilidad. La fusión es completa si toda la información de todos los esquemas se integra en el esquema común.

Una fusión no es mínima cuando la información redundante de una relación se mantiene en un esquema integrado a causa de un fallo en la detección de contención. Esquemas no-minimales también pueden resultar del proceso de traducción, debido a la producción de un esquema intermedio que en sí no es mínimo.

La comprensibilidad es la dimensión final para determinar el mejor esquema. Una vez que todos los elementos se combinan, la reestructuración debería facilitar un esquema comprensible. Puede ser necesario obtener un equilibrio entre minimalidad y comprensibilidad, siempre que la fusión y la reestructuración del esquema ha sido completado [6].

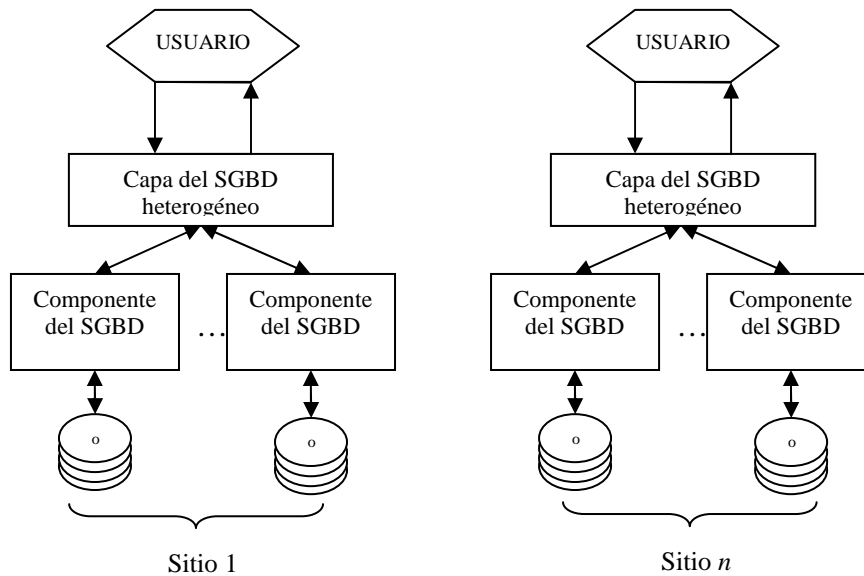
### **2.7.5 Procesamiento de consultas en BBDD heterogéneas.**

La naturaleza de los sistemas que integran varias BBDD requiere pasos ligeramente diferentes a los necesarios en el procesamiento de consultas distribuidas [6] (Véase apartado 2.4).

Cada SGBD tiene sus propios procesadores de consulta y es más complejo por las siguientes razones:

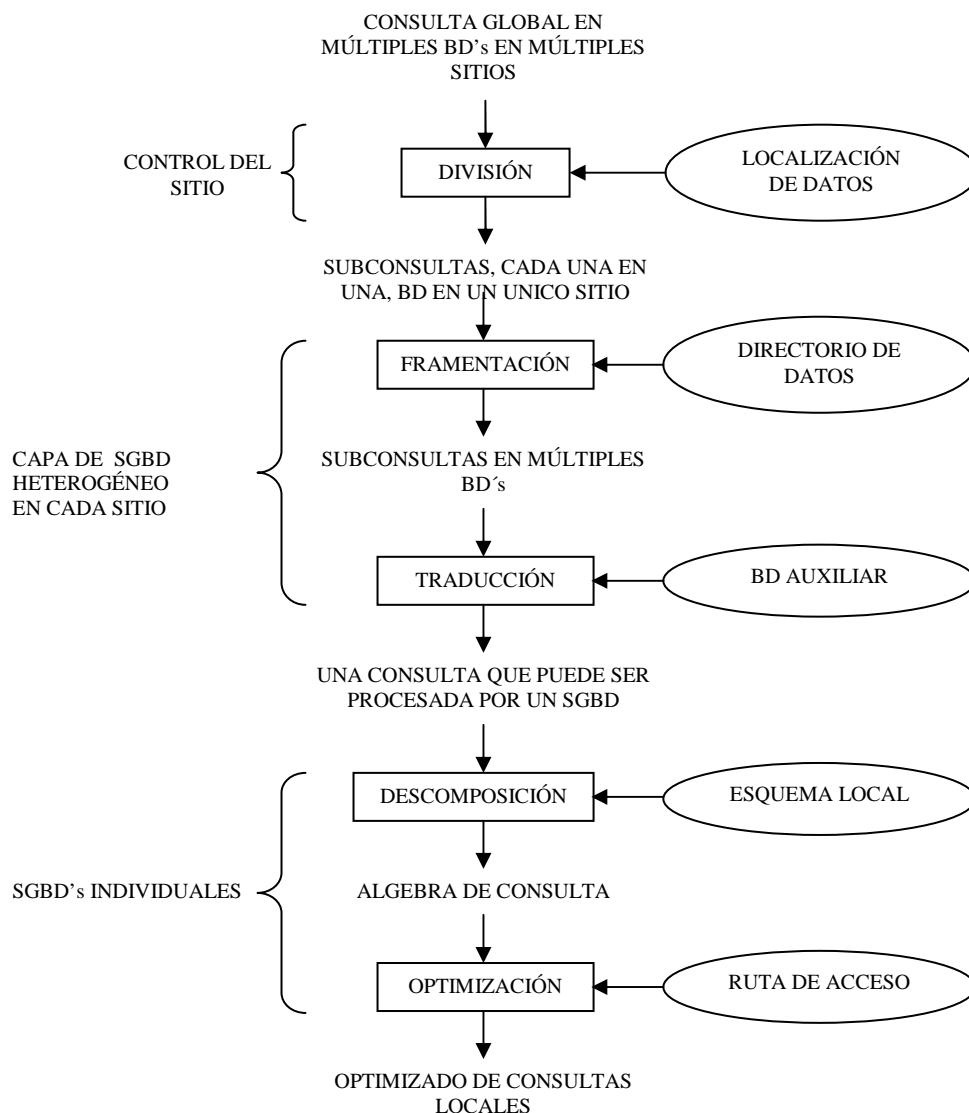
- La capacidad de los componentes del SGBD pueden ser diferentes, lo que impide un tratamiento uniforme de las consultas a través de múltiples SGBD y sitios.
- Del mismo modo, el coste de tramitación de las consultas puede ser diferente en diferentes SGBD. Esto aumenta la complejidad de las funciones de costes que deben ser evaluadas.
- Puede haber dificultades en el movimiento de datos entre SGBDs ya que pueden diferir en su capacidad para leer datos "movidos".
- La capacidad de optimización local de cada SGBD puede ser muy diferente.

La arquitectura representada en la Ilustración 13 muestra una complejidad adicional en los SGBD distribuidos heterogéneos. En los SGBD distribuidos, el procesamiento de consultas tiene que tratar sólo con datos a través de múltiples sitios. En un entorno SGBD distribuido heterogéneo, por otra parte, los datos son distribuidos no sólo a través de los sitios sino también a través de múltiples BBDD, cada una gestionada por un SGBD autónomo. De esta manera, mientras hay dos divisiones que cooperan en el proceso de consultas en un SGBD distribuido (el control del sitio y los sitios locales), el número de divisiones aumenta a tres en el caso de SGBD distribuido heterogéneo:



**Ilustración 13 Estructura de un SGBDD heterogéneo**

- El SGBD heterogéneo, recibe la consulta global. Cuando una consulta es recibida en un nodo, lo primero que necesita hacer es “dividirla” en subconsultas basadas en la distribución de datos a través de múltiples sitios. En este paso, sólo es necesario preocuparse de la ubicación de los datos a través de los sitios, más que de su almacenamiento a través de varias BDs.
- Cada una de las subconsultas se envía al sitio donde va a ser procesada. Es aquí donde cada subconsulta es traducida en el lenguaje del SGBD respectivo. A pesar de que esta información se puede mantener dentro del directorio, es común almacenarlo como una BD auxiliar.
- El componente de los SGBDs, que en última instancia optimiza y ejecuta la consulta. Véase la Ilustración 14.



#### Ilustración 14 Pasos del procesamiento de consultas en Sistemas Multibase

En cuanto la optimización de consultas en varios SGBDs, es similar a la de SGBD distribuidos en algunos aspectos, pero diferente en otros. Al igual que con los SGBDs distribuidos homogéneos, la optimización de consultas en varios SGBDs puede estar basada en la heurística o basada en los costes. Dos heurísticas alternativas se pueden emplear en la descomposición de una consulta en subconsultas:

- La primera alternativa consiste en descomponer una consulta global en el menor número de subconsultas posible, cada una de ellas es ejecutada por un componente del SGBD.

Ventajas: La descomposición es relativamente simple, y hay más oportunidades de optimización en el nivel de consulta optimizador global.

Desventaja: El procesador de consultas a nivel global y el optimizador hace más trabajo, y hay más mensajes que se transmiten para ejecutar la consulta.

- La segunda heurística alternativa es descomponer la consulta global en el mayor número de subconsultas posible, cada una de ellas es ejecutada por un SGBD.

Ventajas: el procesador de consultas a nivel global y el optimizador hace menos trabajo, ya que entre el espacio de procesamiento se reduce al mínimo.

Desventaja: Esto se traduce en menos mensajes, pero en los procesadores de interfaz de componentes más sofisticados (CIPS, *component interface processors*). Los CIPs y el optimizador de componentes SGBD están más involucrados en la transformación y optimización. Los CIPs pueden intercambiar mensajes entre sí, y puede almacenar, borrar, y operar en los archivos temporales.

## **2.7.6 Gestión de transacciones**

El desafío es permitir actualizaciones globales simultáneas de los componentes de BDs sin violar su autonomía. En general, no es posible proporcionar la misma semántica que los SGBDD (homogéneos) sin violar una cierta autonomía [6].

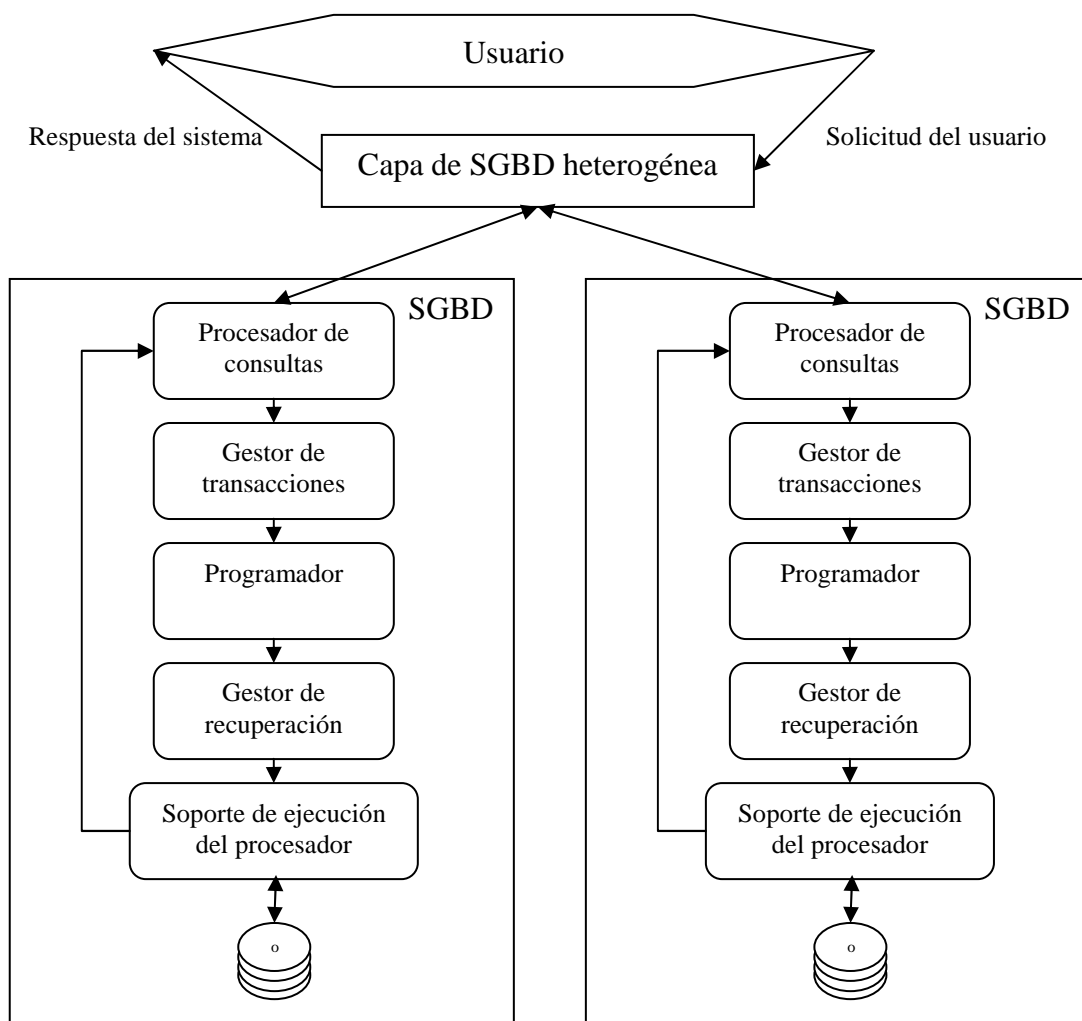
### **2.7.6.1 Modelo de transacción y computación.**

La arquitectura de un SGBD heterogéneo implica una serie de SGBDs, cada uno con su propio administrador de transacciones (llamados gestores locales de transacciones o LTM) y una capa de SGBD heterogénea en la parte superior. El administrador de transacciones de la capa del SGBD heterogénea de varios SGBD se llama administrador de transacciones globales (GTM), ya que gestiona la ejecución de operaciones globales. Un esquema de la arquitectura de un entorno distribuido de SGBD heterogéneo puede verse en la Ilustración 15.

En un sistema heterogéneo, hay dos tipos de operaciones:

- las transacciones locales, que se envían a cada SGBD y se ejecutan en una única BD
- y las transacciones globales, que tienen acceso a múltiples BDs que se envían a la capa de múltiples SGBD. Una transacción global se divide en una serie de subtransacciones globales, cada una de las cuales se ejecuta en una BD.





**Ilustración 15 Componentes de un SGBD heterogéneo [6]**

### 2.7.6.2 Control de concurrencia en sistemas heterogéneos.

Dos transacciones son conflictivas si tienen una operación que accede al mismo dato y al menos una de ellas es de escritura [6]. No es sencillo para el gestor de Transacciones Globales (TG) determinar los conflictos en un sistema heterogéneo. Supóngase que dos transacciones globales que se manejan por el gestor de TG no parecen estar en conflicto en absoluto. Sin embargo, la existencia de transacciones locales puede provocar conflictos en los componentes de BD a través de transacciones, los conflictos indirectos de este tipo no pueden ser detectados por el gestor de TG y son un origen de dificultades significativas en múltiples SGBD.

Una serie de condiciones han sido definidas para especificar cuándo se pueden actualizar las transacciones globales de forma segura en un sistema heterogéneo. Estas condiciones son útiles para determinar la funcionalidad mínima requerida de los diversos gestores de transacciones.

- La primera condición para proporcionar el control de concurrencia global es que los DBA individuales sean responsables de la correcta ejecución de las operaciones en sus respectivas BBDD.
- La segunda condición requiere que cada gestor local de transacciones (LTM) mantenga el orden de ejecución relativa de las subtransacciones determinadas por el GTM. El GTM, entonces, se encarga de coordinar la sumisión de las subtransacciones globales al LTM y la coordinación de su ejecución.
- Por otra parte, el GTM es el responsable de tratar con interbloqueos globales que se producen en las transacciones globales.

## **2.8 Herramientas utilizadas**

En este apartado se van a introducir brevemente las herramientas principales utilizadas para el desarrollo del proyecto.

### **2.8.1 El SGBD Oracle**

Oracle es un SGBD relacional desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- soporte de transacciones,
- estabilidad,
- escalabilidad
- soporte multiplataforma
- herramienta cliente/servidor

Un SGBD relacional Oracle esta compuesto por tres partes principales, que son:

- el kernel de Oracle
- las instancias del sistema de Base de Datos.
- los archivos relacionados al sistema de Base de Datos.

### **2.8.2 El SGBD SQLServer**

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son Transact-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes SGBDs.

Las características de este SGBD son las siguientes:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.

- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

### 2.8.3 Servidor Vinculado

Un servidor vinculado es una herramienta de SQL Server que permite ejecutar comandos en orígenes de datos OLE DB situados en servidores remotos [6]. Los servidores vinculados ofrecen las siguientes ventajas:

- Acceso al servidor remoto.
- Capacidad de ejecutar consultas distribuidas, actualizaciones, comandos y transacciones en orígenes de datos heterogéneos en toda la organización.
- Capacidad de tratar diferentes orígenes de datos de manera similar, como por ejemplo es el caso de SGBDD heterogéneos.

#### Componentes de servidores vinculados

Una definición de servidor vinculado especifica los siguientes objetos:

- Un proveedor OLE DB
- Un origen de datos OLE DB

Un proveedor OLE DB es una biblioteca DLL que administra un origen de datos específico e interactúa con él. Un origen de datos OLE DB identifica la base de datos específica a la que se puede tener acceso mediante OLE DB. Aunque los orígenes de datos en los que se realizan consultas a través de definiciones de servidores vinculados son bases de datos normales, existen proveedores OLE DB para una amplia variedad de archivos y formatos de archivo. Se trata de archivos de texto, datos de hojas de cálculo y los resultados de búsquedas de contenido de texto.

### 2.8.4 Database Link

Un enlace de base de datos es una herramienta de Oracle para conectar a una BD remota a través de una BD local [3]. Los enlaces son útiles cuando se desea consultar una tabla en una BDD, o incluso para insertar en una tabla local datos localizados en otra tabla remota.

Existen tres tipos básicos de enlaces de base de datos: privados, públicos y globales:

- Los enlaces de tipo **global** tienen cobertura sobre toda una red y son los utilizados por servidores de Oracle Names
- Un enlace de tipo **privado** pertenece al usuario que lo crea. La sintaxis de creación es la siguiente:

```
CREATE DATABASE LINK enlace_local
CONNECT TO xe IDENTIFIED BY usuario
USING 'remota.bd'
```

Donde:

- **enlace\_local** es el nombre del enlace que se crea.
- **xe** es la cuenta a la que conectarse, en la base de datos remota (nombre de usuario y contraseña).
- **'remota.bd'** es el nombre global de la base de datos remota.

De esta forma el usuario conectará con la BD remota remota.bd usando el nombre de usuario y contraseña de xen dicha BD. Para poder crear este enlace privado, un usuario debe tener el privilegio de “crear enlace privado a base de datos” en la BD local, así como el de “crear sesión” en la BD remota.

Tras la creación del enlace, el usuario responsable puede consultar las tablas de xe en la base de datos remota remota.bd.

Desde el punto de vista de seguridad, los enlaces privados de BD son preferibles a los enlaces públicos porque los enlaces privados sólo están disponibles para su creador. En general, se debe optar por un enlace de BD privada siempre que sea posible y ver el enlace público como un caso especial o como último recurso. Algunas situaciones específicas que requieren enlaces privados son:

- Los enlaces que se utilizan para actualizaciones instantáneas
- Los enlaces que se utilizan en *triggers*
- Los enlaces que conectan a una cuenta con privilegios en la BD remota
- Ciertas configuraciones de replicación avanzada
- Un enlace **público** permite a cualquier usuario o cualquier programa PL/SQL acceder a objetos en una BD remota. La creación de este tipo de enlaces a BD muy similar a la de los enlaces privados, únicamente se precisa añadir la palabra *public* a la sentencia de creación del enlace.

Los enlaces públicos permiten la conexión a cualquier usuario, incluso si en la BD remota no existe ninguno con su combinación nombre/contraseña, esto es así porque un enlace público permite a cualquier otro usuario emplear el nombre de usuario/contraseña de quién creó el enlace.

Un enlace de BD público es apropiado si muchos usuarios de una aplicación deben tener acceso a un objeto remoto y es irrazonable o imposible la creación de cuentas individuales para cada uno de ellos en la BD remota. En esta situación, el DBA puede crear una sola cuenta en la BD a distancia para que el enlace se conecte a la BD.

En resumen, se hace uso de enlaces privados de BD si es posible y de los públicos si es necesario.

Los enlaces a bases de datos son unidireccionales, para conseguir doble direccionalidad se precisará la creación de dos enlaces.

Para crear un enlace privado de BD, los usuarios deben tener el privilegio de sistema CREATE DATABASE LINK.

Para crear un enlace de BD pública, los usuarios deben tener el privilegio del sistema CREATE PUBLIC DATABASE LINK.

Además, la cuenta desde la que el enlace se conecta a la BD debe tener privilegios de CREATE SESSION.

## **3 Estudio de la Viabilidad y Planteamiento del proyecto**

### ***3.1 Descripción general del sistema y alcance del proyecto***

Dentro de los objetivos del proyecto está el realizar

- el estudio en profundidad de los sistemas de BDD heterogéneas (Véase el apartado 2) tratando de forma teórica el diseño, el control semántico de los datos, el procesamiento de consultas, el control de concurrencia y el control de recuperación de BDD; los SGBD Oracle y SQL Server y sus herramientas de comunicación en entornos de BBDD distribuidas: database link y servidor vinculado respectivamente.
- llevar a cabo un proyecto de diseño de bases de datos distribuidas heterogéneas que abarque las fases de planificación, análisis, diseño, implementación y pruebas, documentando cada una de estas fases.

La fase de implementación se establecerá sobre un determinado supuesto práctico, para lo cual se realizarán las instalaciones y configuraciones de software necesarias para crear el sistema distribuido mencionado anteriormente.

Los SGBD con los que se ha decidido llevar a cabo este desarrollo son Microsoft SQL Server y Oracle por su importancia en el entorno empresarial. Para simular un entorno distribuido se ha instalado cada SGBD en un ordenador diferente, simulando cada uno una sede.

El sistema implementado se inspira en el entorno de la Universidad Carlos III de Madrid y en el desarrollo de algunas de sus actividades, de carácter académico y también económico, en dos de sus campus, el situado en la localidad madrileña de Leganés y el situado en Getafe. Cada uno de estos campus es un nodo de nuestro sistema distribuido que está conectado mediante una red. En cada uno de los campus habrá diferentes departamentos que gestionaran datos de distinta índole.

Dado este supuesto, es asumido que en la sede de Leganés se prevé una carga menor de trabajo mientras que en la sede de Getafe, por ser la sede con mayor estimación de carga, dispondrá de hardware más potente.

La BD será utilizada por dos tipos de usuario en cada sede:

- en la sede de Leganés habrá un tipo de usuario que se encargue de la gestión académica y otro tipo de usuario DBA de Oracle.

- en la sede de Getafe habrá un tipo de usuario que se encargue de la gestión académica y de la gestión de nóminas y otro tipo de usuario DBA de SQL Server.

### 3.2 Planificación temporal del proyecto y estimación de costes

Para realizar la planificación del proyecto se ha utilizado el programa Microsoft Project, resumiéndose a continuación la duración, el coste y los recursos utilizados durante su ejecución.

El proyecto se ha dividido principalmente en tres fases:

- Fase I: Estudio del problema, análisis y diseño
- Fase II: Implementación física: configuración hardware, instalación y conexión entre SGBDs.
- Fase III: Implementación lógica: resolución e implantación de un caso práctico y pruebas de ejecución.

Se puede observar la planificación completa del proyecto en el fichero [Microsoft Project](#) que se entrega adjunto a este documento<sup>2</sup>. No obstante en la Tabla 2 se presenta un resumen de la duración en días y el coste total de los recursos humanos en euros del mismo:

FASE	DURACIÓN	COSTE RRHH
Fase I	90d	14.912,20 €
Fase II	33d	6.969,60 €
Fase III	98d	13.776,44 €
<b>TOTAL</b>	<b>221 días</b>	<b>35.658,24 €</b>

**Tabla 2 Resumen de los costes de personal conforme a las fases del proyecto**

A continuación en la Tabla 3 se hace un desglose de los recursos de personal utilizados, su dedicación en horas y su coste:

NOMBRE DEL RECURSO	DEDICACIÓN	TASA ESTÁNDAR	COSTE
Jefe de proyecto	160 horas	46,21 €/hora	7.393,28 €
Analista de BD	240 horas	36,42 €/hora	8.739,84 €
Diseñador de BD	192 horas	32,00 €/hora	6.144,00 €
DBA	384 horas	26,40 €/hora	10.137,60 €
Programador BD	40 horas	21,10 €/hora	843,52 €
Pruebas BD	100 horas	24,00 €/hora	2.400,00 €
<b>TOTAL</b>			<b>35.658,24 €</b>

**Tabla 3 Resumen del coste de los recursos personales del proyecto**

<sup>2</sup> Ver archivo adjunto a la memoria “[Diseño y Construcción de Bases de Datos Distribuidas Heterogéneas sobre Oracle y SQL Server.mpp](#)”

En la Tabla 4 se hace un desglose de los recursos materiales utilizados, su coste de amortización en años, los años que se prevé estará activo en el proyecto y su coste:

NOMBRE DEL RECURSO	COSTE DE AMORTIZACIÓN ANUAL	AÑOS AMORTIZADOS	COSTE
Servidor de Getafe	1000,00 €	3 años	3000,00 €
Servidor de Leganés	857,14 €	3 años	2571,43 €
Licencia Windows sede Getafe	--	--	127 €
Licencia Windows sede Leganés	--	--	127 €
Licencia de Oracle 10g	--	--	300 €
Licencia SQL Server 2008	--	--	300 €
<b>TOTAL</b>			<b>6.425,43 €</b>

**Tabla 4 Resumen del coste de los recursos materiales del proyecto**

Se puede observar en la Tabla 5 el cálculo del coste de amortización de los recursos materiales utilizados durante el proyecto:

RECURSO	COSTE	DURACIÓN	AMORTIZACIÓN/año
Servidor de Getafe	7000 €	7 años	1000,00 €/año
Servidor de Leganés	6000 €	7 años	857,14 €/año

**Tabla 5 Cálculo del coste amortización de los recursos materiales**

A modo de resumen, en la Tabla 6 se puede apreciar el coste total del proyecto, que se estima en 42.083,67 €, habiéndose previsto su inicio el día Lunes 3 de Mayo de 2010 y su finalización el 20 de Octubre de 2010.

CONCEPTO	COSTE
Costes de recursos personales	35.658,24 €
Costes de recursos materiales	6.425,43 €
<b>TOTAL</b>	<b>42.083,67 €</b>

**Tabla 6 Suma final de los costes materiales y personales**

### 3.3 Supuesto práctico

A continuación se expone el caso práctico elegido para llevar a cabo el análisis, el diseño y la implantación de un sistema de BDD dando autonomía a cada una de las sedes:

*“La Universidad Carlos III tiene en la actualidad dos campus distribuidos en las siguientes poblaciones de la Comunidad de Madrid: Getafe y Leganés. Esta ampliación ha repercutido en los sistemas informáticos de la Universidad llevándose a cabo el tendido de una red para unir los dos campus y ahora se baraja la posibilidad de diseñar una BDD, actualmente centralizada en Getafe e implantada sobre un SGBD SQL Server, para manejar de forma autónoma información sobre las titulaciones ofertadas. Sin embargo la sede de Leganés hace unos meses adquirió unas licencias de Oracle y la implantación de la BD se haría sobre este gestor.*

*En cada uno de los campus se quiere crear un departamento para la gestión de las distintas titulaciones impartidas en dicho campus, además de manejar los datos acerca de los cursos de los que consta cada una de ellas y los grupos que forman parte de estos cursos. En este departamento también se desea guardar información de los profesores (nombre, e-mail y número de despacho) que imparte clases en cada uno de los campus.*

*El departamento de gestión de nóminas y contratación del profesorado se mantendrá en Getafe.”*

## 4 Análisis y Diseño del Sistema

### 4.1 Especificación de requisitos

#### 4.1.1 Requisitos hardware

- Dos servidores con la configuración necesaria para instalar SQL Server 2008 (véase la Tabla 14) y Oracle 10g XE (véase el apartado 5.1.1).
- La red LAN existente que conecta la sede de Leganés y Getafe.

#### 4.1.2 Requisitos software

- Oracle 10g XE
- SQL Server 2008
- Oracle Database 10g Client Release 2
- Oracle 10g Release 2 Oracle Data Access Components

#### 4.1.3 Requisitos de usuario y de sistema

El campo ID hace referencia al identificador del requisito, que debe ser unívoco. El formato a seguir es el siguiente: R<tipo>-<números>. En este caso el campo <tipo> se corresponde con caracteres del nombre de tipo de requisito (las cuales se definen en la Tabla 7 para los requisitos de usuario y la Tabla 8 para los requisitos de sistema) y <números> son números enteros consecutivos incrementados en una unidad.

Tipo de requisito	Acrónimo
Capacidad	CP
Restricción	RS

Tabla 7 Definición de identificadores de tipos de requisitos de usuario

Tipo de requisito	Acrónimo
Funcional	FN
Rendimiento	RD
Interfaz	IN
Operación	OP
Recursos	RC
Comprobación	CM
Documentación	DC
Seguridad	SG

Tabla 8 Definición de identificadores de tipos de requisitos de sistema



#### 4.1.3.1 Requisitos de usuario

A continuación se muestran en la Tabla 9 los requisitos de usuario recogidos durante la fase de análisis del sistema:

ID	Nombre
RCP001	Consulta local de los datos de forma transparente.
RCP002	Actualización local de los datos de forma transparente.
RCP003	Borrado local de los datos de forma transparente.
RCP004	Altas de los datos de forma transparente.
RCP005	Acceso transparente a los objetos y los datos de la sede de Leganés desde la sede de Getafe.
RCP006	Acceso transparente a los objetos y los datos de la sede de Getafe desde la sede de Leganés.
RCP007	Los datos referentes a las nóminas de los profesores se almacenan en la sede de Getafe.
RCP008	Gestionar en cada uno de los campus las distintas titulaciones impartidas en dicho campus.
RCP009	Gestionar los datos acerca de los cursos de los que consta cada una de las titulaciones.
RCP010	Gestionar los datos acerca de los grupos que forman parte de los cursos de los que consta cada una de las titulaciones.
RCP11	Gestionar información de los profesores (nombre, e-mail y número de despacho) que imparte clases en cada uno de los campus.
RRS001	El SGBD para la sede de Leganés será Oracle.
RRS002	El SGBD para la sede de Getafe será SQL Server.
RRS003	Los datos de la tabla PLUSES_HIJO tienen un número bajo de actualizaciones.
RRS004	Los datos de la tabla PLUSES_HIJO requieren una disponibilidad alta.
RRS005	Los datos de la tabla CLASIFICACIONES tienen un número bajo de actualizaciones.
RRS006	Las réplicas para la tabla CLASIFICACIONES requieren una disponibilidad media.
RRS007	La tabla PLUSES_HIJO de la sede de Leganés estará replicada en la sede de Getafe.
RRS008	La tabla CLASIFICACIONES de la sede de Leganés estará replicada en la sede de Getafe.
RRS009	En cada sede habrá x usuarios administradores con todos los permisos sobre la BD.
RRS010	En cada sede habrá x usuarios no administrador con los permisos indispensables para la gestión de los datos.
RRS011	El SGBD para la sede de Leganés será Oracle.
RRS012	El SGBD para la sede de Getafe será SQL Server.
RRS013	Recursos hardware más potentes se instalarán en Getafe.

**Tabla 9 Requisitos de usuario**

#### 4.1.3.2 Requisitos de sistema

A continuación se muestran en la Tabla 10 los requisitos de sistema recogidos durante la fase de análisis del mismo:

ID	Nombre
<b>RFN001</b>	Consulta de los datos de forma transparente.
<b>RFN002</b>	Actualización de los datos de forma transparente.
<b>RFN003</b>	Borrado de los datos de forma transparente.
<b>RFN004</b>	Altas de los datos de forma transparente.
<b>RFN005</b>	Acceso a los objetos y los datos de la sede de Leganés desde la sede de Getafe.
<b>RFN006</b>	Acceso a los objetos y los datos de la sede de Getafe desde la sede de Leganés.
<b>RFN007</b>	Se mantendrán claves primarias en cada tabla.
<b>RFN008</b>	Las tablas estarán relacionadas mediante sus claves primarias y través de sus claves ajenas.
<b>RFN009</b>	El SGBD para la sede de Leganés será Oracle.
<b>RFN010</b>	El SGBD para la sede de Getafe será SQL Server.
<b>RRD001</b>	Disponibilidad del sistema de 24x7 horas.
<b>RRD002</b>	Tabla de “PLUSES_HIJO” replicada en la sede de Getafe.
<b>RRD003</b>	Tabla de “CLASIFICACIONES” replicada en la sede de Getafe.
<b>RRD004</b>	Los datos referentes a las nóminas de los profesores se almacenan en la sede de Getafe.
<b>RRD005</b>	Los datos que no están almacenados en las tablas “CLASIFICACIONES” o “PLUSES_HIJO” serán almacenados en la sede a la que se refiere.
<b>RRD006</b>	Las réplicas para la tabla PLUSES_HIJO serán instantáneas.
<b>RRD007</b>	Las réplicas para la tabla CLASIFICACIONES serán cada 24 horas.
<b>RRD008</b>	La tabla PLUSES_HIJO de la sede de Leganés estará replicada en la sede de Getafe.
<b>RRD009</b>	La tabla CLASIFICACIONES de la sede de Leganés estará replicada en la sede de Getafe.
<b>ROP001</b>	Las claves primarias serán “L-“seguido de un número secuencial creciente en uno para la sede de Leganés.
<b>ROP002</b>	Las claves primarias serán “G-“seguido de un número secuencial creciente en uno para la sede de Getafe.
<b>RRC001</b>	Las sedes se conectan por medio de una LAN.

**Tabla 10 Requisitos de sistema**

## 4.2 Estrategia de diseño de la BDD

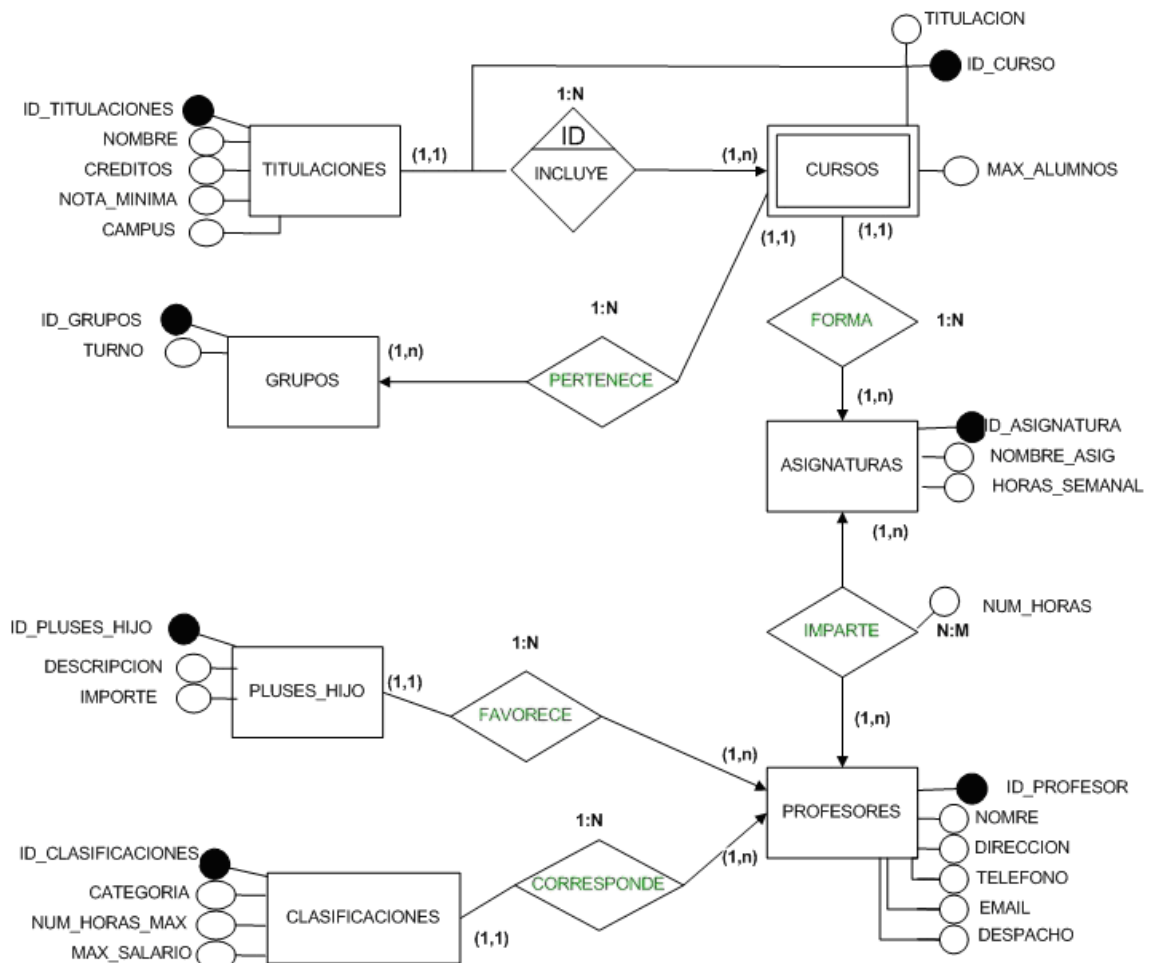
Tras evaluar el planteamiento del caso práctico comienza la etapa de diseño del entorno de distribución. Los pasos llevados a cabo durante la etapa de diseño de la BDD son los siguientes:

### Diseño centralizado

En este apartado se muestra el diagrama entidad relación, y su esquema relacional, y los supuestos adicionales.

#### 4.2.1 Diagrama E/R

El diagrama E/R resultante es el mostrado en la Ilustración 16



**Ilustración 16 Diagrama E/R correspondiente a una BD centralizada para la gestión de la Universidad Carlos III**

#### 4.2.2 Supuestos semánticos asociados al diagrama E/R

A continuación se listan los supuestos semánticos del sistema que no se han podido recoger en el diagrama.

- El campo TURNO de la tabla GRUPOS podrá tomar el valor “TARDE” o “MAÑANA”.
- El campo TELEFONO de la tabla PROFESORES será un número entero de longitud fija 9 y que comience por 6 o por 9.
- El campo EMAIL de la tabla PROFESORES tendrá un formato del tipo <nombre.apellido>@uc3m.es.
- El campo DESPACHO de la tabla PROFESORES será de una longitud fija de 7 caracteres de los cuales tendrán la siguiente estructura N.N.LNN donde N representa un número de 0 a 9 y L representa una letra en mayúsculas. Un de esta estructura sería ejemplo: 2.1.C15
- El campo CLASIFICACIONES(NUM\_HORAS\_MAX) será mayor o igual que la suma del campo IMPARTIR(NUM\_HORAS) de todos los registros de cada profesor de esa categoría.
- Un GRUPO no existirá en más de una titulación.
- El campo IMPORTE de la tabla PLUSES\_HIJO  $\geq 0$

- El campo MAX\_SALARIO de la tabla CLASIFICACIONES  $\geq 0$

### 4.2.3 Supuestos semánticos no incluidos en el enunciado

- IMPARTIR(NUM\_HORAS) es el número de horas que un profesor impartirá clase por cada asignatura
- cada titulación incluye al menos un curso
- un curso es incluido al menos en una titulación
- cada grupo pertenece al menos a un curso
- cada curso tiene al menos un grupo de estudiantes
- cada asignatura forma parte de al menos un curso
- cada curso es formado al menos por una asignatura
- cada asignatura será impartida por al menos un profesor
- un profesor impartirá al menos una asignatura
- una clasificación corresponde al menos a un profesor
- a cada profesor le corresponde una única clasificación
- cada plus\_hijo favorece al menos a un profesor
- cada profesor es favorecido por un plus\_hijo

### 4.2.4 Transformación al esquema relacional

El grafo relacional resultante es el mostrado en la Ilustración 17. En este nuevo grafo que surgirá a partir del esquema E/R, obtenemos una nueva relación por cada entidad, donde su clave primaria se derivará directamente del atributo identificador principal de cada entidad.

En cuanto a las interrelaciones que existen entre entidades, se detalla su transformación a continuación:

#### Interrelaciones 1:N

Se encuentran entre estos casos las interrelaciones:

PERTENECE entre GRUPOS y CURSOS,  
 INCLUYE entre TITULACIONES y CURSOS,  
 FORMA entre CURSOS y ASIGNATURAS,  
 FAVORECE entre PLUSES\_HIJO y PROFESORES y  
 ASIGNADA entre CLASIFICACIONES y PROFESORES

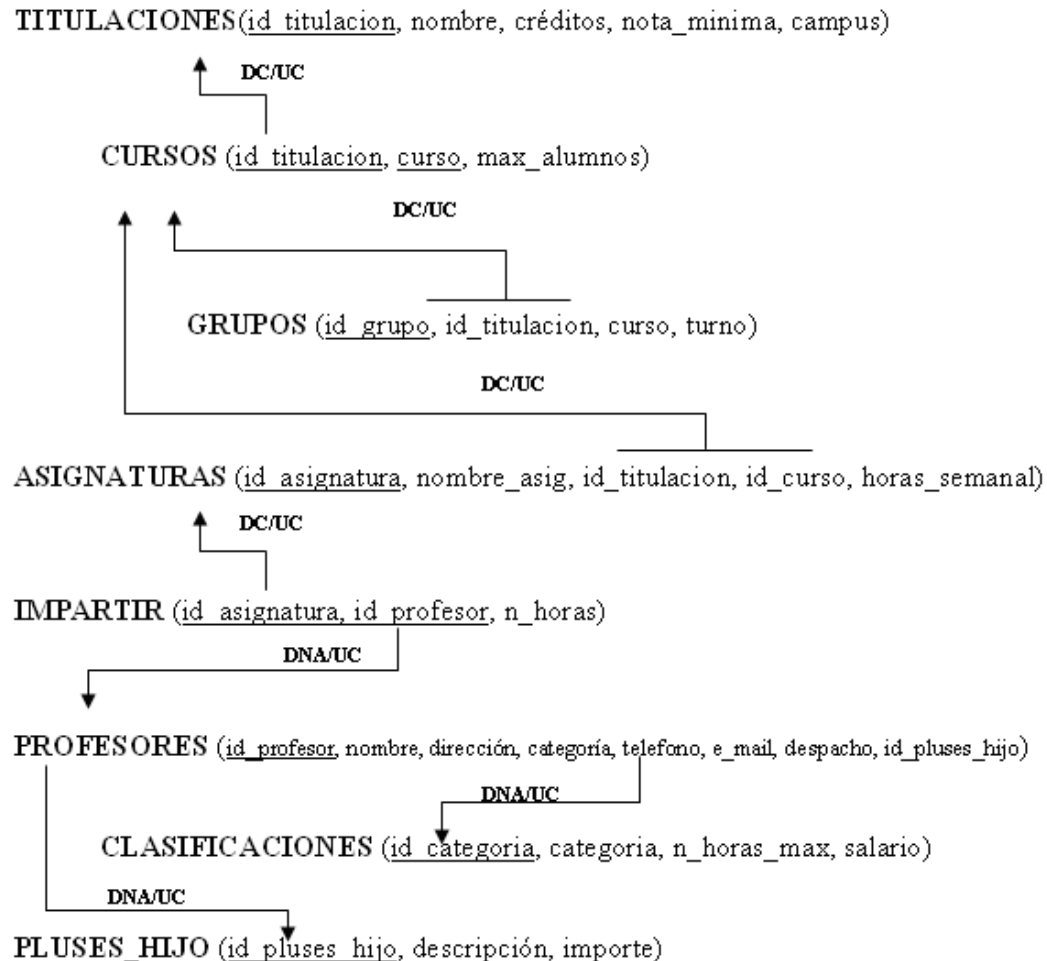
Todas estas relaciones provocan la propagación de clave primaria, como así fue indicado en el esquema E/R (véase la Ilustración 16).

#### Interrelaciones N:M

Estas relaciones se transformarán en una tabla intermedia, cuyas claves primarias serán los identificadores principales de las tablas implicadas en la relación.

Se encuentra en este caso la relación IMPARTE, entre PROFESORES y ASIGNATURAS, que se ha transformado en el grafo en la tabla IMPARTIR.

Respecto a las restricciones de integridad que deben cumplir las tablas, se permite el borrado y modificación en cascada (DC/UC) en todas las tablas que contienen claves foráneas excepto en la tabla intermedia IMPARTIR y la tablas CLASIFICACIONES y PLUSES\_HIJO, en las que se deniega el borrado y se permite la actualización en cascada (DNA/UC). Para borrar en las tablas a las que referencia la tabla IMPARTIR o CLASIFICACIONES y PLUSES\_HIJO, primero habrá que borrar en la propia tabla.



**Ilustración 17** Esquema relacional correspondiente a una BD centralizada para la gestión de la Universidad Carlos III

## Diseño de la distribución

### 4.2.5 Identificación de los nodos de distribución.

El sistema está formado por dos sedes. A una de estas sedes la identificaremos como *Sede de Leganés* y a la otra como *Sede de Getafe*. Ambas nominaciones se realizan en consecuencia a la localización de los campus en los que residen las BBDD. Y los roles del sistema se dividen de la siguiente manera:

- Sede Getafe. Roles
  - Nóminas y contratación de profesores.
  - Gestión de estudios. (cod\_campus=Getafe).

- Sede Leganés. Rol
  - Gestión de estudios (cod\_campus = Leganés), independencia local de cada centro con respecto a sus datos locales.

#### 4.2.6 Asignación inicial de tablas y nodos.

A la vista de los requerimientos del sistema y de la autonomía local que se quiere dar a los datos, se prevé almacenar la información de las tablas IMPARTIR, GRUPOS, ASIGNATURAS, CURSOS, INFO\_PROFESORES, CLASIFICACIONES, PLUSES\_HIJO y TITULACIONES referentes al campus de Getafe en la sede de Getafe y los referentes al campus de Leganés en la sede de Leganés. Es decir, todos los datos se almacenarán en la sede de la que procedan, a excepción de los datos de las nóminas que se almacenarán en Getafe porque el rol correspondiente se desarrolla sólo en dicha sede. Véase la Tabla 11 en la que se muestra la correspondencia entre la tabla que almacena los datos y el campus al que se refieren estos.

TABLA	CAMPUS DE PROCEDENCIA DATOS	SEDE ALMACENAMIENTO
Impartir	Getafe	Getafe
	Leganés	Leganés
Grupos	Getafe	Getafe
	Leganés	Leganés
Asignaturas	Getafe	Getafe
	Leganés	Leganés
Cursos	Getafe	Getafe
	Leganés	Leganés
Info_profesores	Getafe	Getafe
	Leganés	Leganés
Clasificaciones	Getafe	Getafe
	Leganés	Leganés
Pluses_hijo	Getafe	Getafe
	Leganés	Leganés
Titulaciones	Getafe	Getafe
	Leganés	Leganés
Nominas_profesores	Getafe	Getafe
	Leganés	Getafe

Tabla 11 Asignación inicial de tablas y nodos.

#### 4.2.7 Análisis de los criterios de fragmentación.

Dado que el criterio de fragmentación del supuesto es dar autonomía funcional a los distintos campus, se supondrá una fragmentación en la que cada sede almacene los datos como se especificó en el apartado 4.2.6. Véanse los pasos seguidos junto con su motivación:

Se quiere que en cada campus se maneje la información de las titulaciones que allí se imparten. Por ello se aplica una fragmentación horizontal primaria en la relación TITULACIONES.

TITULACIONES\_i:  $\sigma_{\text{campus} = 'i'}$  (TITULACIONES)

Donde  $i = \{\text{Getafe, Leganés, Colmenarejo}\}$

Para tener la información completa sobre las titulaciones de cada campus necesitamos conocer qué cursos se imparten y los grupos formados para cada curso, por lo que tendremos, tanto para la relación CURSOS como para la relación GRUPOS, que realizar una fragmentación horizontal derivada.

$\text{CURSOS}_i: \text{CURSOS} \xrightarrow{\text{id\_titulacion}} \text{TITULACIONES}_i$

$\text{GRUPOS}_i: \text{GRUPO} \xrightarrow{\text{id\_titulacion, curso}} \text{CURSOS}_i$   
Donde  $i = \{\text{Getafe, Leganés}\}$

De la misma forma asignamos las asignaturas impartidas en las distintas titulaciones a los campus correspondientes realizando una fragmentación horizontal derivada de la relación de asignaturas con relación a los fragmentos los cursos correspondientes a cada campus ( $\text{CURSOS}_i$ )

$\text{ASIGNATURAS}_i: \text{ASIGNATURAS} \xrightarrow{\text{id\_titulacion, curso}} \text{CURSOS}_i$   
Donde  $i = \{\text{Getafe, Leganés}\}$

Por último para conocer los profesores que dan clases en los distintos campus realizamos otra fragmentación derivada de la relación que almacena la impartición docente de los profesores con respecto a los fragmentos de las asignaturas correspondientes a cada campus

$\text{IMPARTIR}_i: \text{IMPARTIR} \xrightarrow{\text{id\_asignatura}} \text{ASIGNATURAS}_i$   
Donde  $i = \{\text{Getafe, Leganés}\}$

Para mantener en Getafe los datos de gestión de nóminas y contratación del profesorado, primero se realiza una fragmentación vertical para ubicar en dicha sede sólo la información que interesa.

$\text{INFO\_PROFESORES}: \prod \text{id\_profesor, nombre, e-mail, despacho} (\text{PROFESORES})$

$\text{NOMINAS\_PROFESORES}: \prod \text{id\_profesor, direccion, telefono, categoria, id\_pluses\_hijo} (\text{PROFESORES})$

Para luego realizar una fragmentación horizontal derivada y de esta forma obtener fragmentos con los atributos de los profesores necesarios en cada uno de los campus.

$\text{INFO\_PROFESORES}_i = \text{id\_asignatura} \xrightarrow{\text{IMPARTIR}_i}$   
Donde  $i = \{\text{Getafe, Leganés}\}$

Las relaciones CLASIFICACIONES y PLUSES\_HIJO no se fragmentarán por contener información relevante en ambos campus. De esta manera los datos originales se

ubicarán en una sede y se barajará posteriormente la conveniencia de su replicación en la otra sede para ahorrar accesos remotos.

#### 4.2.8 Asignación de fragmentos a las sedes.

A continuación, en la Tabla 12 se muestra la asignación de cada uno de los fragmentos obtenidos en el apartado 4.2.7.

Relación	Getafe	Leganés
<b>TITULACIONES</b>	TITULACIONES_Getafe	TITULACIONES_Leganés
<b>CURSOS</b>	CURSOS_Getafe	CURSOS_Leganés
<b>GRUPOS</b>	GRUPOS_Getafe	GRUPOS_Leganés
<b>ASIGNATURAS</b>	ASIGNATURAS_Getafe	ASIGNATURAS_Leganés
<b>IMPARTIR</b>	IMPARTIR_Getafe	IMPARTIR_Leganés
<b>CLASIFICACIONES</b>	CLASIFICACIONES	
<b>PLUSES_HIJO</b>	PLUSES_HIJO	
<b>PROFESORES</b>	INFO_PROFESORES_Getafe	INFO_PROFESORES_Leganés
	NOMINAS_PROFESORES	

Tabla 12 Esquema de asignación

#### 4.2.9 Análisis de replicación

En el supuesto a tratar, las relaciones CLASIFICACIONES y PLUSES\_HIJO tendrán pocas actualizaciones, por lo que se van a replicar en todas las sedes. La información contenida en la tabla PLUSES\_HIJO es necesario mantenerla actualizada y por esto las replicaciones serán instantáneas. En cambio la información de la tabla CLASIFICACIONES no es tan elemental y las replicaciones se realizarán todos los días a la misma hora, a las 24:00.

#### 4.2.10 Esquema de replicación

A continuación, en la Tabla 13 se pueden ver las relaciones replicadas que se indicaron en el apartado 4.2.9.

Relación	Getafe	Leganés
CLASIFICACIONES	RG_CLASIFICACIONES	RL_CLASIFICACIONES
PLUSES_HIJO	RG_PLUSES_HIJO	RL_PLUSES_HIJO

Tabla 13 Esquema de replicación

### 4.3 Diseño y descripción de la arquitectura

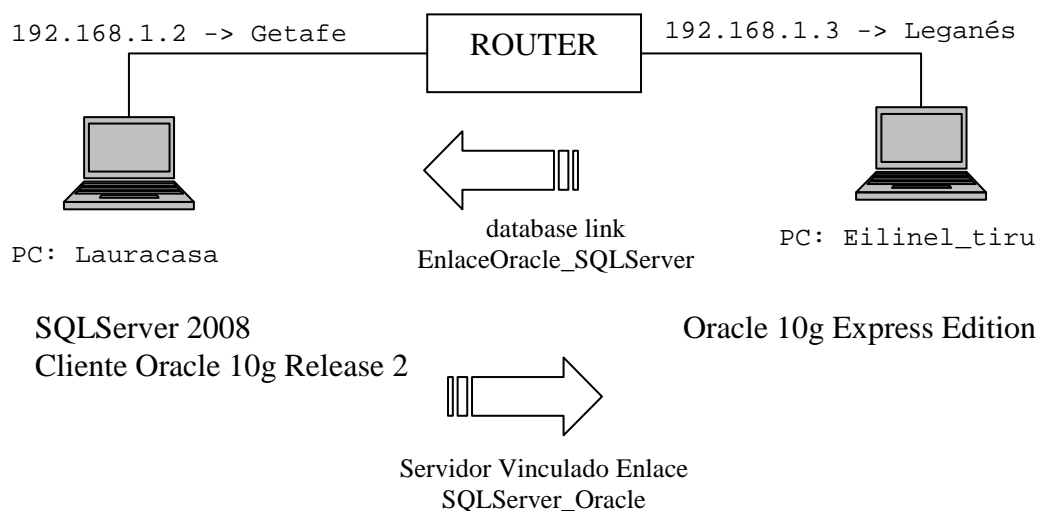
Para este supuesto se ha decidido llevar a cabo un entorno con dos sedes físicas, “Leganés” y “Getafe”, en las que como se aprecia en la Ilustración 18 se ha instalado el software SQL Server 2008 y Cliente Oracle 10g Release 2 en una sede y Oracle 10g Express Edition en otra, respectivamente. Esta decisión se toma con el fin de poder implementar un entorno con SGBD heterogéneos. El sistema operativo de ambos equipos es “Windows XP Professional”. Ambos nodos están conectados mediante un



router configurado de tal manera que a nivel de red permite la distribución de los datos de las aplicaciones de BD.

La sede de Getafe está instalada en el servidor con nombre de equipo *Lauracasa* y la sede de Leganés está instalada en el servidor con nombre de equipo *Eilinel\_tiru*.

Ambos equipos están conectados bajo una red de área local (LAN), en la que el equipo *Lauracasa* es referenciado con la IP fija 192.168.1.2 y en el equipo *Eilinel\_tiru* con la IP fija 192.168.1.3.



**Ilustración 18 Esquema de la arquitectura del sistema**

Las BBDD pueden ser consultadas en ambos sentidos gracias a:

- la herramienta **Database Link** en un sentido
- la herramienta **Servidor vinculado** en el otro.

Esto es necesario porque los SGBD actuales solo proveen de este tipo de herramientas para conexiones unidireccionales.

### 4.3.1 Sede Leganés

En esta sede se van a crear los siguientes objetos:

- Un usuario con permisos de administrador, llamado **dba\_leganes**.
- Un usuario con todos los permisos necesarios para iniciar una sesión en Oracle y con todos los permisos necesarios para tener recursos para la creación de objetos, llamado **usuario\_leganes**.
- Un tablespace llamado **UC3M** donde se crearan:
  - o las tablas indicadas en la Tabla 12 correspondientes a la sede de Leganés y la tabla PROFESORES\_AUX:
    - IMPARTIR\_LEGANES.
    - GRUPOS\_LEGANES.
    - ASIGNATURAS\_LEGANES.
    - CURSOS\_LEGANES.
    - INFO\_PROFESORES\_LEGANES.

- PROFESORES\_AUX, tabla necesaria para mantener la transparencia de distribución en la fragmentación vertical de la tabla profesores.
  - CLASIFICACIONES.
  - PLUSES\_HIJO.
  - TITULACIONES\_LEGANES.
- Vistas para conseguir consultas transparentes a la BD completa
  - V\_ASIGNATURAS para recuperar todas las tuplas de la tabla ASIGNATURAS de la sede Leganés y Getafe.
  - V\_CLASIFICACIONES para mantener una nomenclatura similar en toda la BD, ya que es una tabla replicada.
  - V\_CURSOS para recuperar todas las tuplas de la tabla CURSOS de la sede Leganés y Getafe.
  - V\_GRUPOS para recuperar todas las tuplas de la tabla GRUPOS de la sede Leganés y Getafe.
  - V\_IMPARTIR para recuperar todas las tuplas de la tabla IMPARTIR de la sede Leganés y Getafe.
  - V\_PROFESORES para recuperar todas las tuplas de la tabla PROFESORES de la sede Leganés y Getafe.
  - V\_TITULACIONES para recuperar todas las tuplas de la tabla TITULACIONES de la sede Leganés y Getafe.
  - V\_PLUSES\_HIJO para mantener una nomenclatura similar en toda la BD, ya que es una tabla replicada.
- Secuencias para mantener las claves primarias de todas las tablas a excepción de INFO\_PROFESORES que dependerá de PROFESORES\_AUX.
  - SQ\_TITULACIONES\_LEGANES;
  - SQ\_GRUPOS\_LEGANES;
  - SQ\_ASIGNATURA\_LEGANES;
  - SQ\_PROFESORES\_LEGANES;
  - SQ\_CLASIFICACIONES\_LEGANES;
- Trigger que asigna la secuencia y le añade el literal “L-” delante del número de secuencia para poder así identificar la sede.
  - inc\_TITULACIONES\_L.
  - inc\_GRUPOS\_L.
  - inc\_ASIGNATURA\_L.
  - inc\_PROFESORES\_L.
  - inc\_CLASIFICACIONES\_L.
  - inc\_PLUSES\_HIJO\_L.
- Enlace público para conectar con la sede de Getafe.
  - ENLACE\_LEGANES\_GETAFE.
- Sinónimos
  - TITULACIONES para TITULACIONES\_LEGANES
  - CURSOS para CURSOS\_LEGANES
  - GRUPOS para GRUPOS\_LEGANES
  - ASIGNATURAS para ASIGNATURAS\_LEGANES
  - IMPARTIR para IMPARTIR\_LEGANES
  - PROFESORES para PROFESORES\_AUX
- Triggers para mantener la transparencia de la fragmentación vertical de la tabla profesores.

- fragmenta\_PROFESORES\_AUX: El usuario realizará una inserción en la tabla PROFESORES y este Trigger se encargará de fragmentar y situar cada uno de los fragmentos en la sede correspondiente.
- actualiza\_PROFESORES\_AUX: El usuario realizará una actualización en la tabla PROFESORES y este Trigger se encargará de actualizar y situar cada uno de los fragmentos en la sede correspondiente.
- borra\_PROFESORES\_AUX: El usuario realizará un borrado en la tabla PROFESORES y este Trigger se encargará de borrar los registros que afecte en la sede que corresponda.
- Trigger para las actualizaciones transparentes. El usuario realizará una actualización sobre cualquiera de las tablas.
  - actualiza\_IMPARTIR.
  - actualiza\_GRUPOS.
  - actualiza\_ASIGNATURAS.
  - actualiza\_CURSOS.
  - actualiza\_CLASIFICACIONES.
  - actualiza\_PLUSES\_HIJO.
  - actualiza\_TITULACIONES.
- Trigger para los borrados transparentes. El usuario realizará un borrado sobre cualquiera de las tablas.
  - borra\_IMPARTIR.
  - borra\_GRUPOS.
  - borra\_ASIGNATURAS.
  - borra\_CURSOS.
  - borra\_CLASIFICACIONES.
  - borra\_PLUSES\_HIJO.
  - borra\_TITULACIONES.
- Vista materializada para realizar la replicación diaria de la tabla CLASIFICACIONES.
- Trigger para realizar la replicación instantánea de la tabla PLUSES\_HIJO.

### 4.3.2 Sede Getafe

En esta sede se van a crear los siguientes objetos:

- Un usuario con permisos de administrador, llamado **dba\_getafe**
- Un usuario con todos los permisos necesarios para iniciar una sesión en SQL Server y con todos los permisos necesarios para tener recursos para la creación de objetos, llamado **usuario\_getafe**.
- Una instancia de BD llamada **UC3M** donde se crearan:
  - las tablas indicadas en la Tabla 12 correspondientes a la sede de Getafe y la tabla PROFESORES\_AUX y TABLA\_SECUENCIAS:
    - IMPARTIR\_GETAFE.
    - GRUPOS\_GETAFE.
    - ASIGNATURAS\_GETAFE.
    - CURSOS\_GETAFE.
    - INFO\_PROFESORES\_GETAFE.
    - NOMINAS\_PROFESORES.

- PROFESORES\_AUX, tabla necesaria para mantener la transparencia de la fragmentación vertical de la tabla profesores.
- R\_CLASIFICACIONES.
- R\_PLUSES\_HIJO.
- TITULACIONES\_GETAFE.
- TABLA\_SECUENCIAS, tabla necesaria para mantener ID's autonumérico que se añadirán como clave primaria al resto de las tablas, simulando el funcionamiento del objeto SEQUENCE en Oracle.
- Vistas para conseguir consultas transparente a la BD completa
  - V\_ASIGNATURAS para recuperar todas las tuplas de la tabla ASIGNATURAS de la sede Leganés y Getafe.
  - V\_CLASIFICACIONES para mantener una nomenclatura similar en toda la BD, ya que es una tabla replicada.
  - V\_CURSOS para recuperar todas las tuplas de la tabla CURSOS de la sede Leganés y Getafe.
  - V\_GRUPOS para recuperar todas las tuplas de la tabla GRUPOS de la sede Leganés y Getafe.
  - V\_IMPARTIR para recuperar todas las tuplas de la tabla IMPARTIR de la sede Leganés y Getafe.
  - V\_PROFESORES para recuperar todas las tuplas de la tabla PROFESORES de la sede Leganés y Getafe.
  - V\_TITULACIONES para recuperar todas las tuplas de la tabla TITULACIONES de la sede Leganés y Getafe.
  - V\_PLUSES\_HIJO para mantener una nomenclatura similar en toda la BD, ya que es una tabla replicada.
- Trigger que asigna la secuencia leída desde la tabla TABLA\_SECUENCIAS, le añade el literal "G-" delante del número de secuencia para poder así identificar la sede e inserta un nuevo registro en la tabla TABLA\_SECUENCIAS para incrementar la "secuencia".
  - inc\_TITULACIONES\_G.
  - inc\_GRUPOS\_G.
  - inc\_ASIGNATURA\_G.
  - inc\_PROFESORES\_G.
  - inc\_CLASIFICACIONES\_G.
  - inc\_PLUSES\_HIJO\_G.
- Servidor Vinculado para conectar con la sede de Leganés.
  - ENLACE\_GETAFE \_ LEGANES.
- Sinónimos
  - TITULACIONES para TITULACIONES\_GETAFE
  - CURSOS para CURSOS\_GETAFE
  - GRUPOS para GRUPOS\_GETAFE
  - ASIGNATURAS para ASIGNATURAS\_GETAFE
  - IMPARTIR para IMPARTIR\_GETAFE
  - PROFESORES para PROFESORES\_AUX
- Triggers para mantener la transparencia de la fragmentación vertical de la tabla profesores.
  - fragmenta\_PROFESORES\_AUX: El usuario realizará una inserción en la tabla PROFESORES y este Trigger se encargará de fragmentar y situar cada uno de los fragmentos en la sede correspondiente.

- actualiza\_PROFESORES\_AUX: El usuario realizará una actualización en la tabla PROFESORES y este Trigger se encargará de actualizar y situar cada uno de los fragmentos en la sede correspondiente.
- borra\_PROFESORES\_AUX: El usuario realizará un borrado en la tabla PROFESORES y este Trigger se encargará de borrar los registros que afecte en la sede que corresponda.
- Trigger para las actualizaciones transparentes. El usuario realizará una actualización sobre cualquiera de las tablas.
  - actualiza\_IMPARTIR.
  - actualiza\_GRUPOS.
  - actualiza\_ASIGNATURAS.
  - actualiza\_CURSOS.
  - actualiza\_CLASIFICACIONES.
  - actualiza\_PLUSES\_HIJO.
  - actualiza\_TITULACIONES.
- Trigger para los borrados transparentes. El usuario realizará un borrado sobre cualquiera de las tablas.
  - borra\_IMPARTIR.
  - borra\_GRUPOS.
  - borra\_ASIGNATURAS.
  - borra\_CURSOS.
  - borra\_CLASIFICACIONES.
  - borra\_PLUSES\_HIJO.
  - borra\_TITULACIONES.

## 5 Implementación del Sistema

### 5.1 Instalación y configuración del servidor Oracle – Sede Leganés

En la sede de Leganés se ha decidido instalar el servidor Oracle por sus características de hardware, es decir, dado que va a ser la sede con menos carga de trabajo se ha optado por instalarla en el equipo con el hardware menos potente.

#### 5.1.1 Instalación Oracle 10g Express Edition

Se realiza una instalación típica del gestor. Para consultar el proceso llevado a cabo paso a paso se puede consultar el 11.1.

#### 5.1.2 Configuración del Servidor Oracle

Se creará un tablespace para almacenar el esquema propuesto y los usuarios que lo utilizarán de la siguiente manera:

En una máquina Windows XP se accederá al servidor Oracle a la herramienta **Comandos SQL**

Para crear el tablespace UC3M, con tamaño inicial de 1024 Mb, y auto extensible

```
CREATE TABLESPACE "UC3M" LOGGING  
DATAFILE 'UC3M.dbf' SIZE 1024M  
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
```

La creación de este tablespace no es obligatoria, pero sí recomendable, así se dispondrá de un espacio propio de datos, evitando interferir con otros ya existentes en el sistema.

- Creación del **usuario administrador**, “dba\_leganes” y contraseña “pfc” que va a trabajar sobre el tablespace “UC3M”, y que será el propietario de los objetos que se creen en el

```
CREATE USER dba_leganes IDENTIFIED BY pfc  
DEFAULT TABLESPACE UC3M  
TEMPORARY TABLESPACE TEMP;
```

Si no se especifica un tablespace, la BD le asignará el tablespace USERS, que es el tablespace que se utiliza por defecto para los nuevos usuarios. Se puede apreciar también que no se ha creado un tablespace para los índices. Si se quisieran mantener datos e índices separados habría que crear otro tablespace y acordarse de especificar éste en las sentencias de creación de índices de este usuario, si no se crearán en UC3M.

Le asignamos los permisos necesarios para trabajar. Si se le asignan los roles “Connect” y “Resource” ya tiene los permisos mínimos, con los que podrá conectarse y realizar las operaciones más habituales de consulta, modificación y creación de objetos en su propio esquema.

```
GRANT "CONNECT" TO "dba_leganes";  
  
GRANT "RESOURCE" TO "dba_leganes";
```

Además se le asignan permisos de administrador.

```
GRANT "DBA" TO "dba_leganes";
```

Para otorgar privilegios sobre objetos específicos sería necesario modificar el usuario con sentencias similares a, que le otorga al usuario permisos para crear secuencias.

```
GRANT CREATE ANY SEQUENCE TO "usu_leganes";
```

- Creación del **usuario**, “usuario\_leganes” con contraseña “pfc” de la misma manera otorgándole tan solo los permisos “Connect” y “Resource”.

### 5.1.3 Otras configuraciones

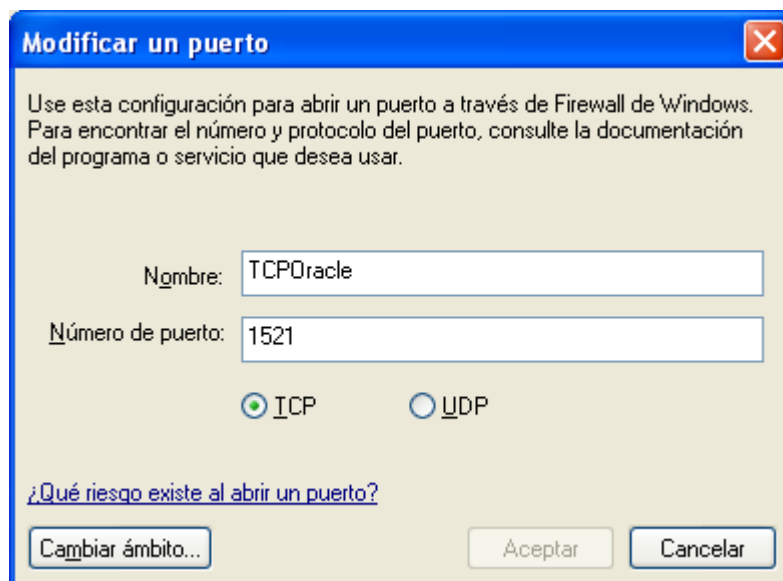
En este apartado se van a comentar una serie de configuraciones necesarias para compartir datos entre dos equipos que no son específicas de los SGBD sino más bien del sistema operativo.

#### 5.1.3.1 Firewall de Windows

Para evitar posibles problemas de conexión entre las dos BBDD por el firewall de Windows se han abierto los puertos tanto UDP como TCP, para Oracle el puerto por defecto es el 1521 y para SQL Server es el 1433 y serán los que se mantendrán en el desarrollo de este proyecto.

Veamos el caso del puerto TCP 1521:

Se accederá al Firewall de Windows de la siguiente manera **Panel de Control >> Firewall de Windows** y en la pestaña **Excepciones** pulsamos **Agregar puerto** y nos aparece una pantalla como puede verse en la Ilustración 19.



**Ilustración 19 Abrir puerto de Firewall de Windows.**

Así creamos una excepción para el firewall de Windows llamada “TCPOracle” para el puerto TCP 1521.

#### 5.1.3.2 Añadir dominios

El archivo *hosts* de un ordenador es usado por el sistema operativo para guardar la correspondencia entre dominios y direcciones IP.

El archivo *hosts* es un archivo de texto plano que puede ser editado por el administrador del equipo. Este archivo es tradicionalmente llamado *hosts* y su ubicación depende del

sistema operativo. En nuestro caso se encuentra en “C:\WINDOWS\system32\drivers\etc”

Añadimos dos líneas que le mostrarán a nuestro sistema operativo qué dirección IP resuelve una dirección de host expresada por su nombre de dominio.

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# Éste es un ejemplo de archivo HOSTS usado por Microsoft TCP/IP para Windows.
# Ejemplos
# 102.54.94.97 rhino.acme.com # servidor origen
# 38.25.63.10 x.acme.com # host cliente x

127.0.0.1 localhost
192.168.1.3 Eilinel_tiru #Sede Leganés
192.168.1.2 Lauracasa #Sede Getafe
```

## ***5.2 Instalación y configuración del Cliente Oracle y del servidor SQLServer – Sede Getafe***

En la sede de Getafe se ha decidido instalar el Cliente Oracle, así como el servidor SQL Server.

Es necesario instalar el siguiente software:

### **5.2.1 Instalación Oracle Database 10g Client Release 2**

Esta herramienta de Oracle se ha de instalar para disponer de las librerías necesarias en los equipos que necesitan tener acceso a un servidor de Oracle (que en nuestro caso es la sede de Leganés).

Para realizar la instalación, se descarga de la Web de Oracle y se descomprime el fichero 10201\_client\_win32.zip. Se ejecuta el fichero “autorun.exe” y se sigue una instalación típica del cliente Oracle como tipo de instalación “Administrador”:

Tras la instalación se podrá realizar la configuración de red de Oracle. Este paso es necesario para poder conectar con el servidor remoto, pero en este caso se hará más adelante configurándolo de forma manual mediante la modificación del fichero tnsnames.ora

### **5.2.2 Instalación de SQL Server 2008**

La versión de SQL Server que se ha instalado es “Microsoft SQL Server 2008 de evaluación.”



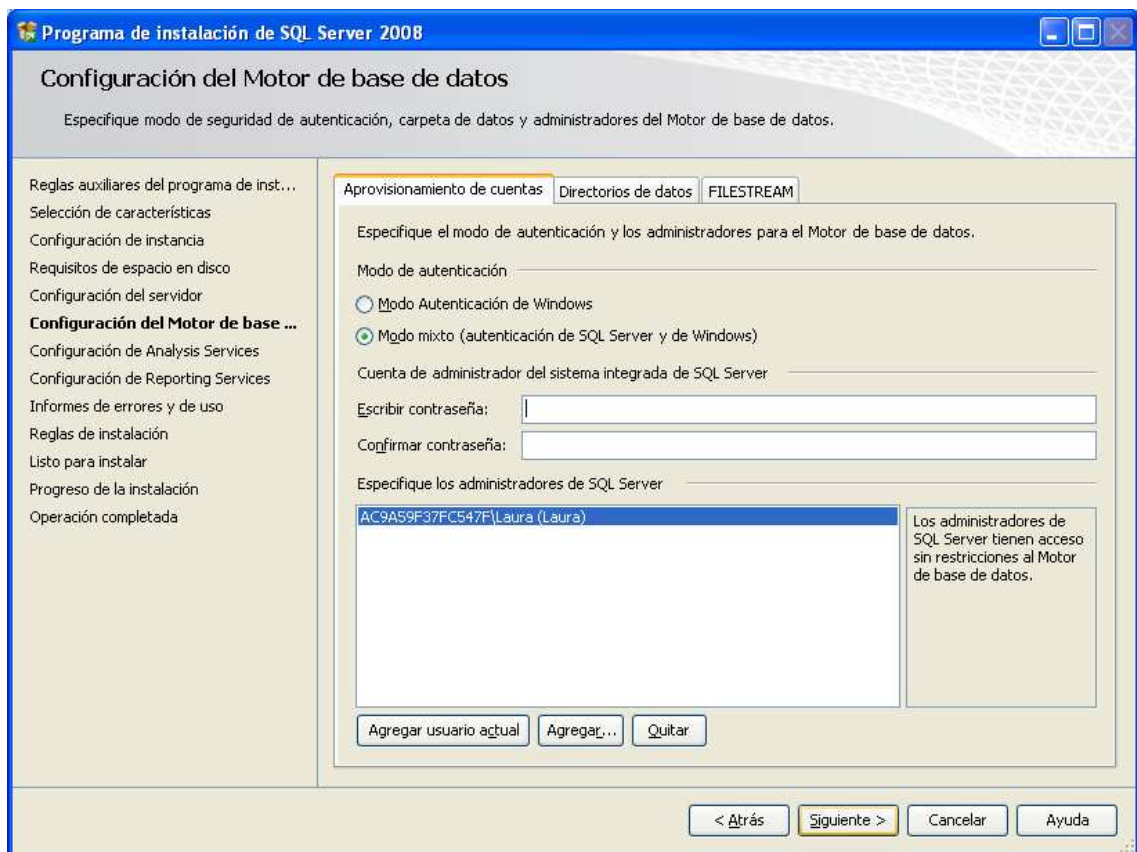
A continuación se presenta el conjunto de requerimientos hardware y software para instalar SQL Server:

Componente	Requisito
Marco de trabajo	<p>El programa de instalación de SQL Server instala los siguientes componentes de software requeridos por el producto:</p> <ul style="list-style-type: none"><li>• .NET Framework 3.5 SP1</li><li>• SQL Server Native Client</li><li>• Archivos auxiliares para la instalación de SQL Server</li></ul>
Software	<p>El programa de instalación de SQL Server requiere Microsoft Windows Installer 4.5 o una versión posterior</p> <p>Una vez instalados los componentes requeridos, el programa de instalación de SQL Server comprobará que el equipo en el que se ha instalado SQL Server 2008 también cumple los demás requisitos para su correcta instalación.</p>
Software de red	<p>Los requisitos de software de red para las versiones de 64 bits de SQL Server 2008 son los mismos que para las versiones de 32 bits.</p> <p>Los sistemas operativos compatibles tienen el software de red integrado. Las instancias predeterminadas y con nombre independientes admiten los siguientes protocolos de red:</p> <ul style="list-style-type: none"><li>• Memoria compartida</li><li>• Canalizaciones con nombre</li><li>• TCP/IP</li><li>• VIA</li></ul>
Disco duro	<p>Las necesidades de espacio en disco variarán con los componentes de SQL Server 2008 que instale.</p>

Unidad	Para la instalación desde disco se necesita una unidad de CD o DVD.
Pantalla	Las herramientas gráficas de SQL Server 2008 requieren VGA o una resolución mayor: resolución mínima de 1.024 x 768 píxeles.
Otros dispositivos	Dispositivo señalador: se necesita un ratón Microsoft o dispositivo señalador compatible.

**Tabla 14 Requerimientos hardware y software para instalar SQL Server**

Se ha llevado a cabo una instalación típica de SQL Server, con una modificación de lo que aparece por defecto en la Ilustración 20: en el apartado de “Modo de autenticación” se ha cambiado de “Modo de Autenticación de Windows” por “Modo mixto (autenticación de SQL Server y de Windows)” dado que la conexión no se llevará a cabo en modo local y de otro modo podría ocasionar problemas a la hora de hacer las conexiones a SQL Server desde Oracle.



**Ilustración 20 Instalación de SQL Server 2008: Aprovisionamiento de cuentas.**

Para consultar el proceso llevado a cabo paso a paso se puede consultar el 11.2

### 5.2.3 Instalación Oracle 10g Release 2 Oracle Data Access Components (ODAC)

El driver OraOLEDB es necesario para hacer una conexión a una BD y más concretamente a una BD Oracle y por ello también para configurar un “Servidor vinculado”. Una vez descargado y descomprimido el fichero de instalación, ejecutamos el fichero setup.exe.

A continuación se detallan los pasos mas relevantes del proceso de instalación de esta herramienta.

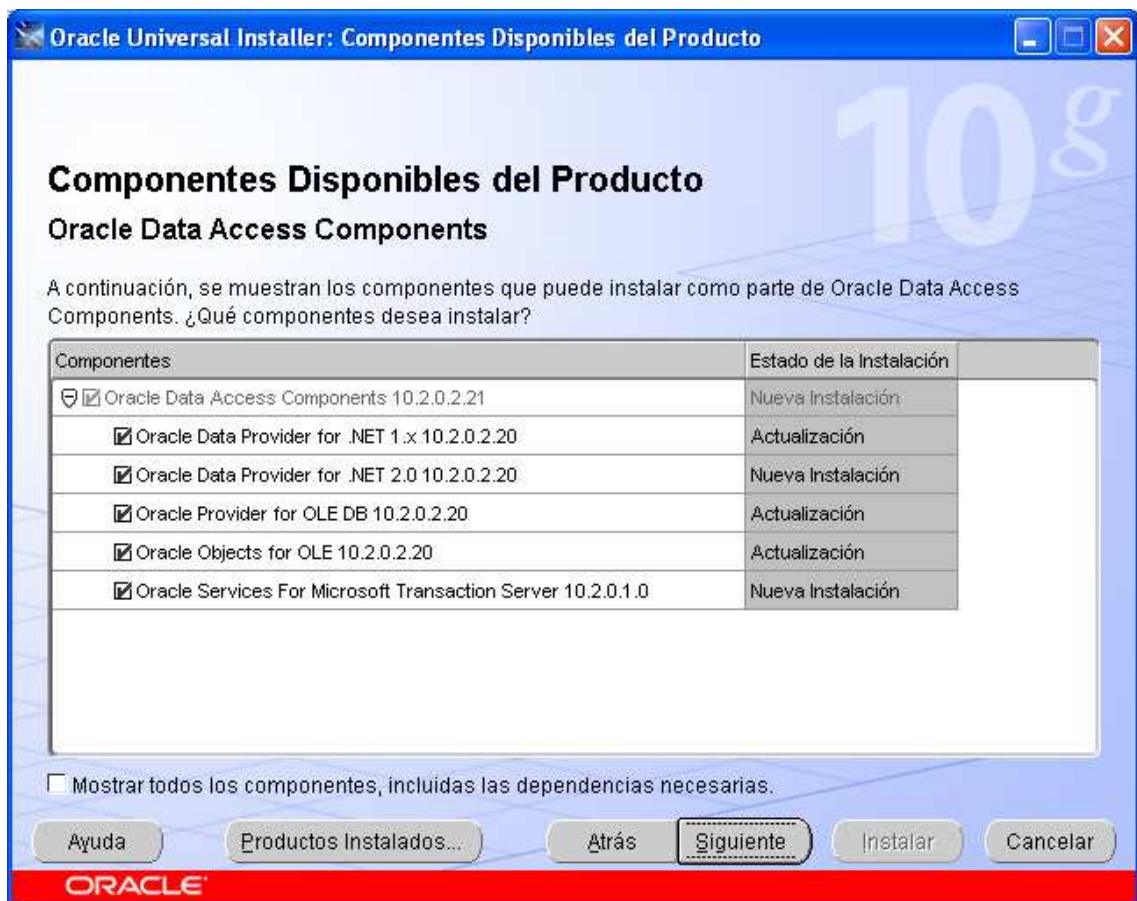
En la pantalla que muestra la Ilustración 21 se selecciona la opción Oracle Data Access Components que incluye el driver OraOLEDB.Oracle



**Ilustración 21 Oracle Universal Installer: Seleccionar producto a instalar**

Hay que seleccionar la instalación de Oracle existente sobre la que se quiere instalar la herramienta.

Aparecen los componentes que se pueden observar en la Ilustración 22 siguientes que pueden ser instalados. Se han seleccionado todos dado que son extremadamente ligeros y pueden ser necesarios en un futuro aunque en principio sólo necesitamos el tercer y cuarto componente.



**Ilustración 22 Oracle Universal Installer: Componentes Disponibles del Producto**

Dado que hemos seleccionado el componente “Oracle Services for Microsoft Transaction Server 10.2.0.1.0”, se pide el número de Puerto que se desea utilizar. El puerto por defecto es el 2030 y se mantiene.

#### 5.2.4 Configuración del servidor SQL Server

Se creará una BD para almacenar el esquema propuesto y los usuarios que lo utilizarán de la siguiente manera:

Para crear la BD UC3M y seleccionarla escribimos el siguiente comando:

```
CREATE DATABASE UC3M;
USE UC3M;
```

- Creación del **usuario administrador**, “dba\_getafe” y contraseña “pfc” que va a trabajar sobre la BD “UC3M”, y que será el propietario de los objetos que se creen en él

```
CREATE USER dba_getafe IDENTIFIED BY 'pfc';
```

Previamente con el comando use UC3M, hemos indicado que todos los objetos que se creen se guardaran en esta BD.

Le asignamos todos los permisos dado que será nuestro usuario administrador.

```
GRANT ALL to dba_getafe;
```

Para otorgar privilegios sobre objetos específicos sería necesario modificar el usuario con sentencias similares a:

```
GRANT privilegios [(columnas)] [, privilegios [(columnas)]] ...  
    ON [objeto] {tabla | * | *.* | basedatos.*}  
    TO usuario [IDENTIFIED BY [PASSWORD] 'contraseña'] [,  
usuario [IDENTIFIED BY [PASSWORD] 'contraseña']] ...  
    [REQUIRE NONE | [{SSL| X509}] [CIPHER 'cipher' [AND]]  
[ISSUER 'issuer' [AND]] [SUBJECT 'subject']]  
    [WITH opcion [opcion] ...]
```

- Creación del **usuario**, “usuario\_getafe” con contraseña “pfc” de la misma manera otorgándole tan solo los permisos “select”, “update”, “insert” y “delete”.

### 5.2.5 Otras configuraciones

En esta sede también es necesario configurar el Firewall y el fichero *hosts* de Windows (Véase 5.1.3).

## 5.3 Configuración del cliente y el servidor para crear el entorno distribuido e interconectar las sedes.

A continuación se procede a explicar los pasos necesarios para conectar las sedes. En el caso de SQL Server es necesario crear un **Servidor Vinculado** hacia Oracle y en el caso de Oracle es necesario crear un **Database Link** hacia SQL Server.

### 5.3.1 Descripción y requisitos para conectar Oracle a SQL Server

Dentro de este apartado se detallarán los pasos llevados a cabo para conectar a la BDD y poder ver los objetos de la sede de Getafe desde la sede de Leganés:

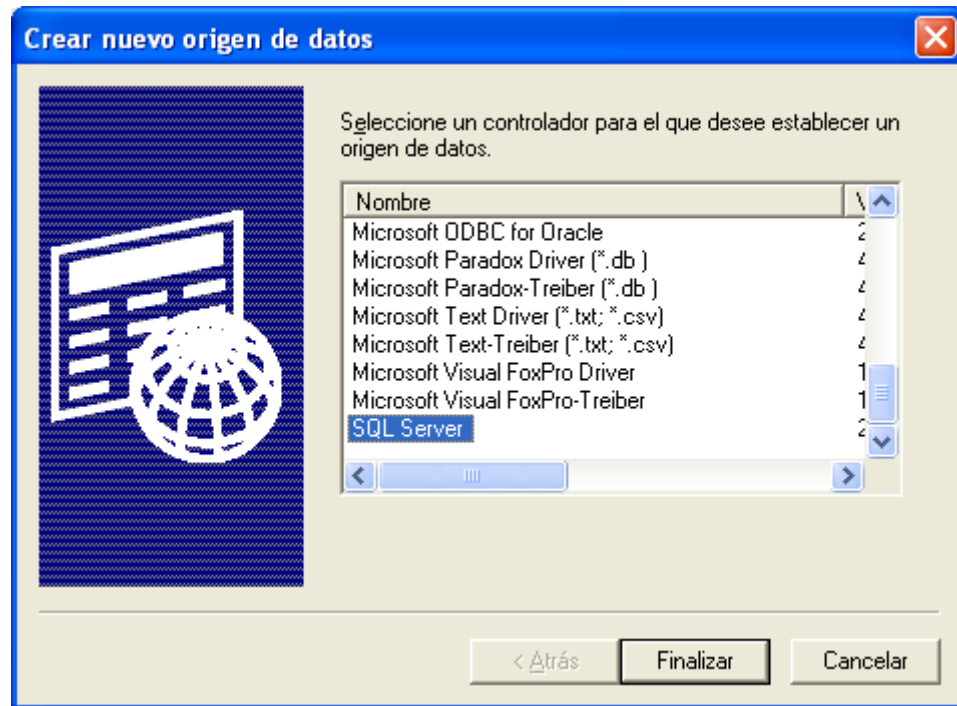
1. Se van a configurar los servicios de conexión heterogéneos de Oracle para poder visualizar BBDD SQL Server desde un esquema Oracle, como si fueran objetos propios de Oracle.

Para poder realizar la configuración es necesario contar con los objetos necesarios del catálogo. Por defecto están instalados, pero en determinadas instalaciones puede ser necesario realizarlo manualmente. Para ello, ejecutar

como usuario SYSTEM el fichero **caths.sql** del directorio “<ORACLE\_HOME>/rdbms/admin”

2. Crear en la máquina servidora de Oracle, un nuevo origen de datos ODBC de SQL Server (DSN de sistema), que apunte a la BD de SQL Server en la sede de Getafe.

Se añade un nuevo origen de datos completando una pantalla similar a la Ilustración 23:



**Ilustración 23 Crear nuevo origen de datos.**

Es necesario seleccionar un controlador para establecer un origen de datos de SQL Server y completar una pantalla similar a la que aparece en la Ilustración 24.



Ilustración 24 Configuración DSN de Microsoft SQL Server 1/3

Los datos que se solicitan son los siguientes:

**Nombre:** es el nombre por el que nos referiremos al origen de datos que estamos creando. (Necesario para establecer el enlace como Oracle).

**Descripción:** es la descripción del origen de datos que estamos creando.

**Servidor:** es el servidor al que nos vamos a conectar, donde está situada la BD. En este caso, la sede de Getafe.

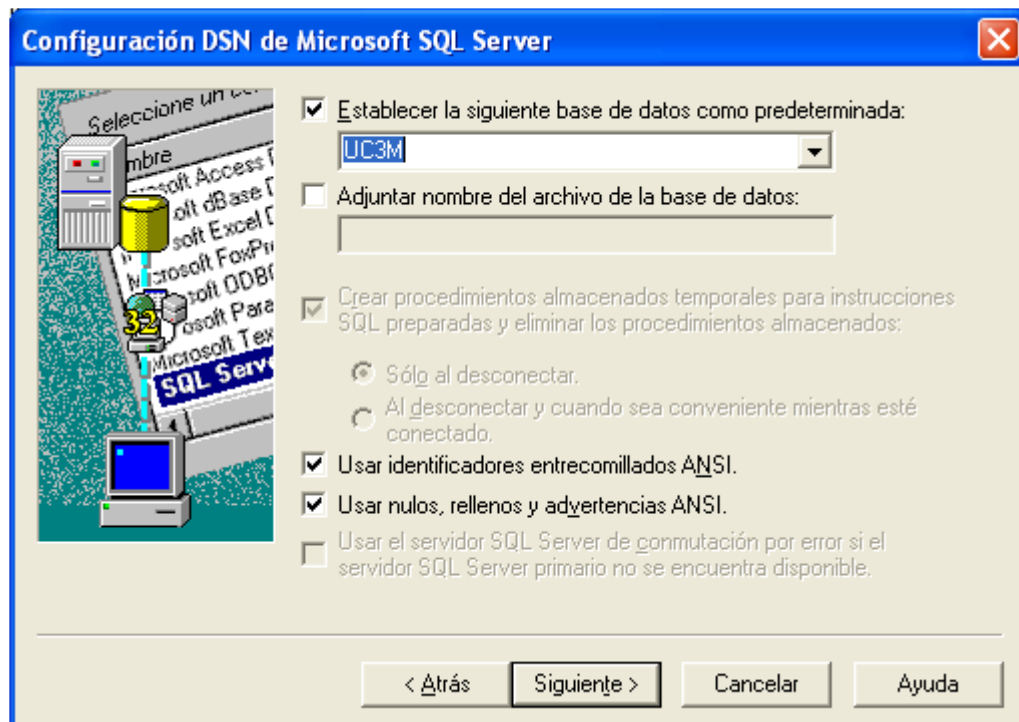
En la pantalla de la Ilustración 25 el tipo de identificación seleccionada se hace excluyendo la opción de conectar con las credenciales del usuario que inició la sesión de Windows. La conexión se realizará, por los motivos explicados anteriormente, mediante una cuenta de usuario remota de Oracle.





**Ilustración 25 Configuración DSN de Microsoft SQL Server 2/3**

A continuación en la Ilustración 26, se selecciona la BD UC3M que se creó en SQL Server y es donde se encuentran los datos a los que necesitamos acceder.



**Ilustración 26 Configuración DSN de Microsoft SQL Server 3/3**

En el resto de pantallas se mantienen las opciones por defecto. Al finalizar se realiza una pruebas satisfactoria de conexión.



3. A continuación es necesario editar el fichero “tnsnames.ora” del servidor de Oracle (sede de Leganés), es posible editarlo con cualquier editor de texto como podría ser el bloc de notas de Windows. Este fichero se encuentra en “<ORACLE\_HOME>/NETWORK/ADMIN” al que se le añaden las siguientes líneas:

```
DSNSQLSERVER=
(DESCRIPTION =
  (ADDRESS_LIST =
    (PROTOCOL = TCP)(HOST = 192.168.1.3)(PORT = 1521)
  )
  (CONNECT_DATA =
    (SERVICE_NAME = DSNSQLServer)
  )
  (HS=OK)
)
```

Explicuemos lo que significa cada campo:

- **DSNSQLSERVER**": nombre y cadena de conexión que se utilizará para el enlace de Oracle con SQLServer.
  - **"HOST=192.168.1.3"**: en el parámetro "HOST" indicaremos el nombre (hostname) o la IP del equipo con la BD Oracle Database, en nuestro caso la sede de Leganés.
  - **"SERVICE\_NAME=DSNSQLSERVER"**: es importante en este parámetro indicar el nombre que le hemos dado al origen de datos (ODBC) en el paso anterior.
  - **"HS = OK"**: con esto le estamos indicando a Oracle que para esta conexión utilice su servicio **Heterogeneous Services** o **Oracle Transparent Gateway**. De esta forma, Oracle tratará esta conexión mediante el origen de datos ODBC indicado.
- a. Desde el servidor Oracle (Sede de Leganés) se crea el **Database Link**: para esto accederemos a Oracle Database con cualquier software que permita ejecutar sentencias SQL de creación de objetos en Oracle. En nuestro caso utilizaremos la herramienta que se instala con el SGBD y a la cual podemos acceder como se muestra en la Ilustración 39.

Ejecutaremos la siguiente sentencia SQL con un usuario con permisos de administrador o en su defecto de creación de DBLINK (GRANT CREATE DATABASE LINK):

```
CREATE PUBLIC DATABASE LINK "ENLACE_Leganes_Getafe"  
CONNECT TO "sa" IDENTIFIED BY "pfc" USING 'DSNSQLServer';  
commit;
```

### 5.3.2 Descripción y requisitos para conectar SQL Server a Oracle

En este apartado se describirán los pasos llevados a cabo para crear un servidor vinculado desde SQL Server a Oracle.

1. Instalar Oracle Database 10g Client Release 2 (Véase apartado 5.2.1)
2. Instalar Oracle 10g Release 2 ODAC (Véase apartado 5.2.3)
3. Configurar una conexión o servicio de red. Para ello es necesario editar el fichero TNSNAMES.ora en la sede de Getafe en la ruta “<ORACLE\_HOME>/product/10.2.0/client\_1/NETWORK/ADMIN/tnsnames.ora” y añadirle las siguientes líneas:

```
XE =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = Eilinel_tiru)(PORT =  
1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = XE)  
    )  
  )
```

Explicuemos lo que significa cada campo:

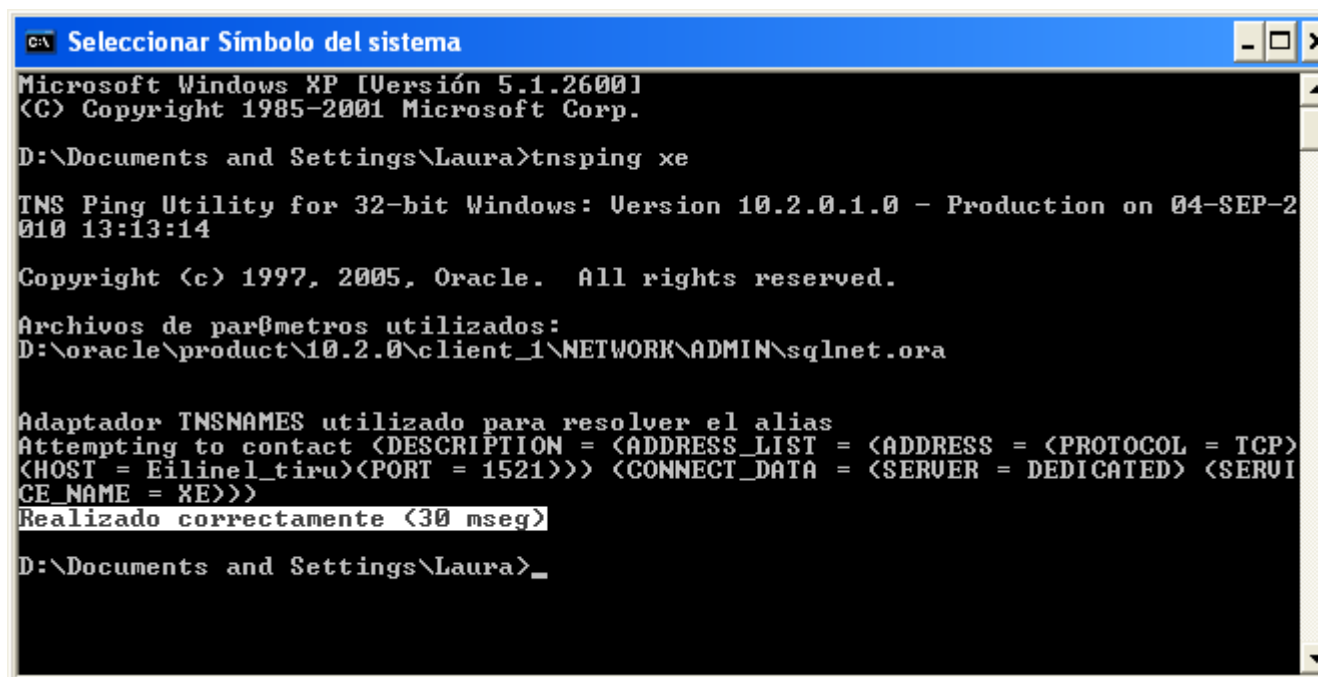
**HOST:** en el parámetro "HOST" indicaremos el nombre (*hostname*) o la IP del equipo con la base de datos Oracle Database, en nuestro caso el equipo “Eilinel\_tiru”.

**PORT:** Puerto donde escucha la base de datos.

**SERVICE\_NAME:** Nombre del servicio de base de datos al que queremos conectar.

**DESCRIPTOR DE CONEXION:** En este caso el descriptor para conectarnos es XE.

Podemos comprobar si la conexión es correcta ejecutando el comando **TNSPING** desde la consola de comandos de Windows de la sede de Getafe como se puede observar en la Ilustración 27:



```
C:\ Seleccionar Símbolo del sistema
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\Documents and Settings\Laura>tnsping xe

TNS Ping Utility for 32-bit Windows: Version 10.2.0.1.0 - Production on 04-SEP-2010 13:13:14

Copyright (c) 1997, 2005, Oracle. All rights reserved.

Archivos de parámetros utilizados:
D:\oracle\product\10.2.0\client_1\NETWORK\ADMIN\sqlnet.ora

Adaptador TNSNAMES utilizado para resolver el alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)
(HOST = Eilinel_tiru)(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = XE)))
Realizado correctamente (30 mseg)

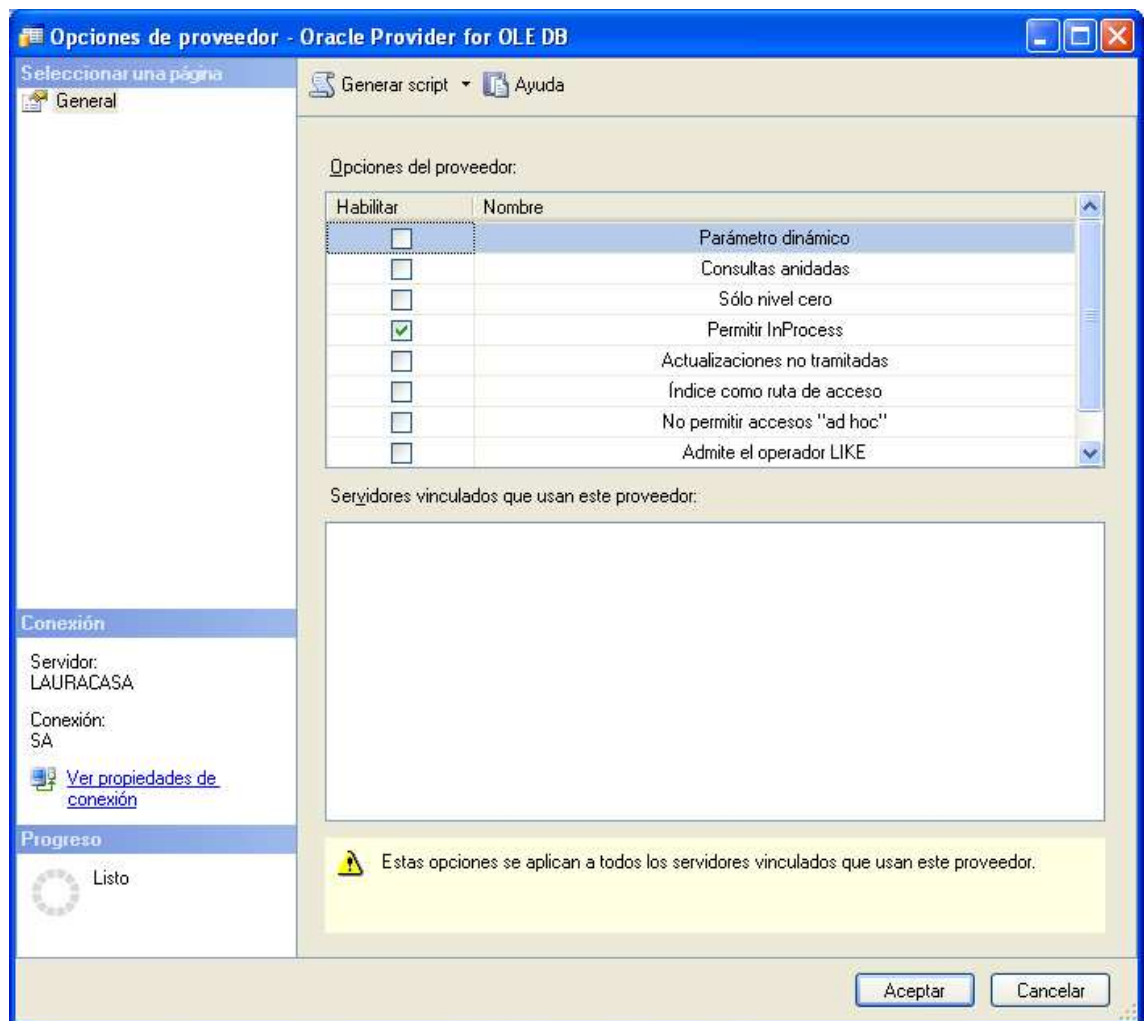
D:\Documents and Settings\Laura>
```

**Ilustración 27 Prueba de conexión de XE**

4. Una vez creado el servicio de red, será necesario añadir la variable de entorno de sistema TNS\_ADMIN con la ruta del archivo tnsnames.ora. Sin ésta el OLE DB/ODBC es incapaz de encontrar la ruta hacia el servicio Oracle.
5. Para que los cambios sean efectivos es necesario reiniciar el servicio de SQL Server desde la consola “Sql Server Configuration Manager”.

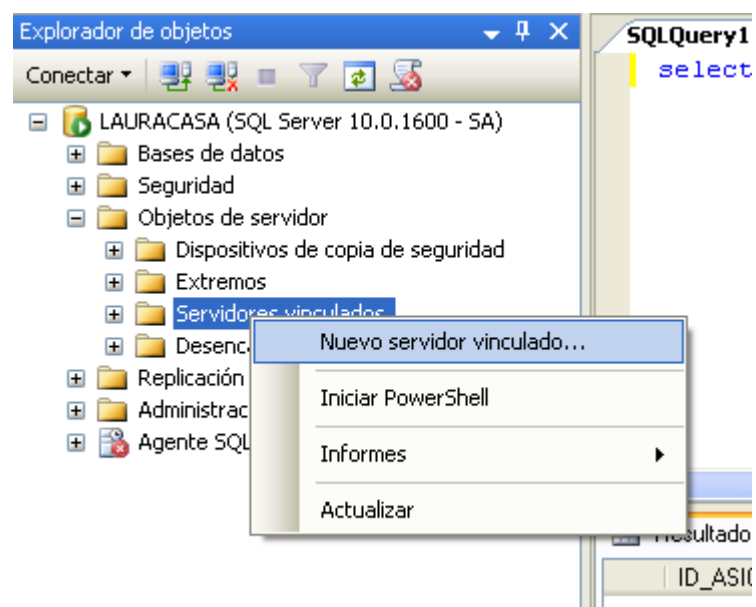
Se configura el proveedor OraOLEDB.Oracle. Para realizar nuestra conexión a la sede de Leganés con Oracle, es necesario seleccionar OraOLEDB.Oracle y marcar la opción “Permitir InProcess”. Véase la Ilustración 28.

SQL Server permite crear una instancia del proveedor como un servidor en proceso. Si no se establece esta opción, el comportamiento predeterminado consiste en crear una instancia del proveedor fuera del proceso de SQL Server. La creación de instancias del proveedor fuera del proceso de SQL Server protege el proceso de SQL Server de posibles errores en el proveedor. Si se crea una instancia del proveedor fuera del proceso de SQL Server, no se permitirán actualizaciones ni inserciones que hagan referencia a columnas long (text, ntext o image) [8].



**Ilustración 28 Opciones de proveedor - Oracle Provider for OLE Db.**

Crear el servidor Vinculado como se puede ver en la Ilustración 29.



**Ilustración 29 Nuevo Servidor Vinculado**

Nos aparece una pantalla similar a la que aparece en la Ilustración 30, por defecto con la opción “General” del menú izquierdo.

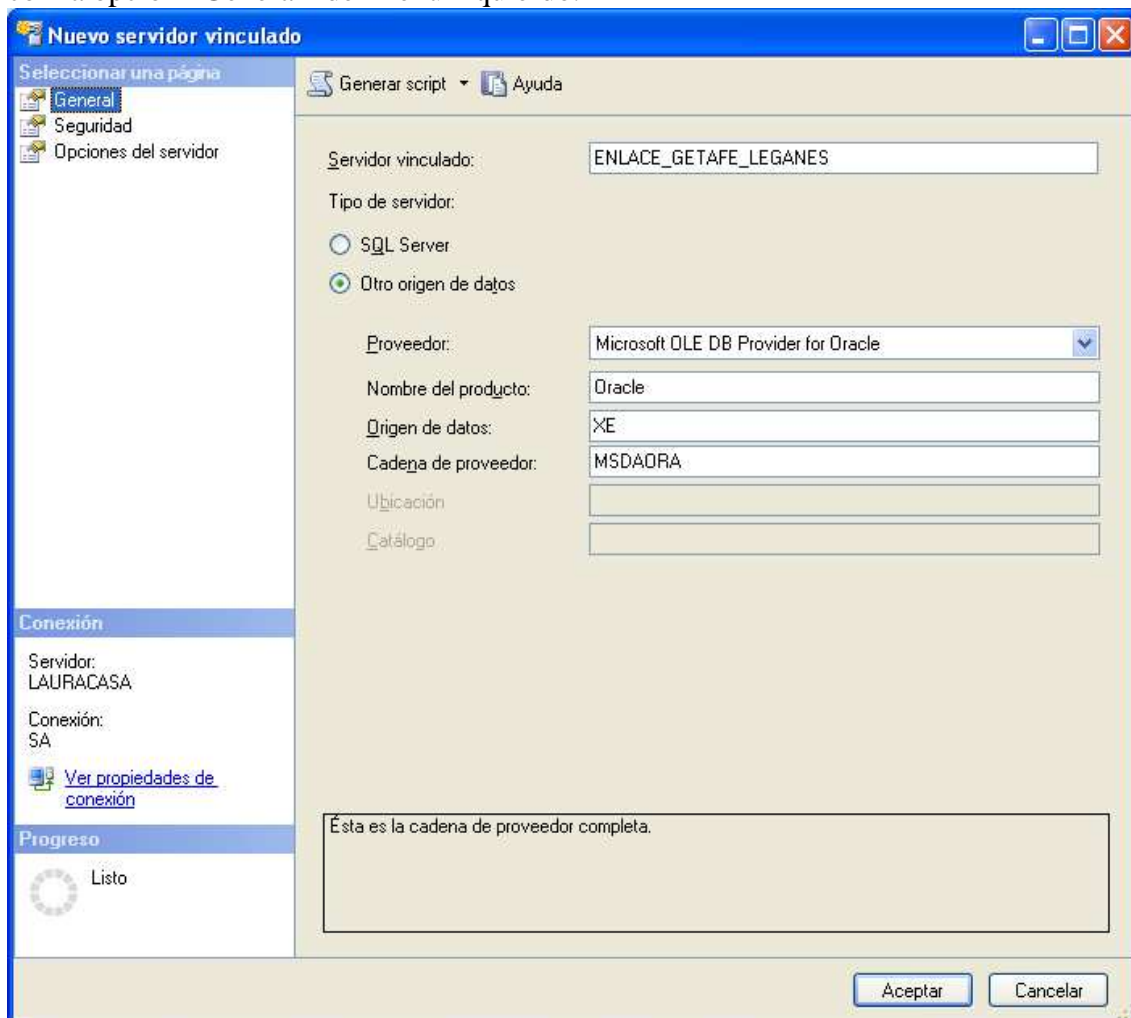


Ilustración 30 Nuevo Servidor Vinculado - general.

Donde se dan los siguientes valores:

**SERVIDOR VINCULADO:** Es el nombre del servidor vinculado que vamos a crear. En nuestro caso “ENLACE\_GETAFE\_LEGANES”.

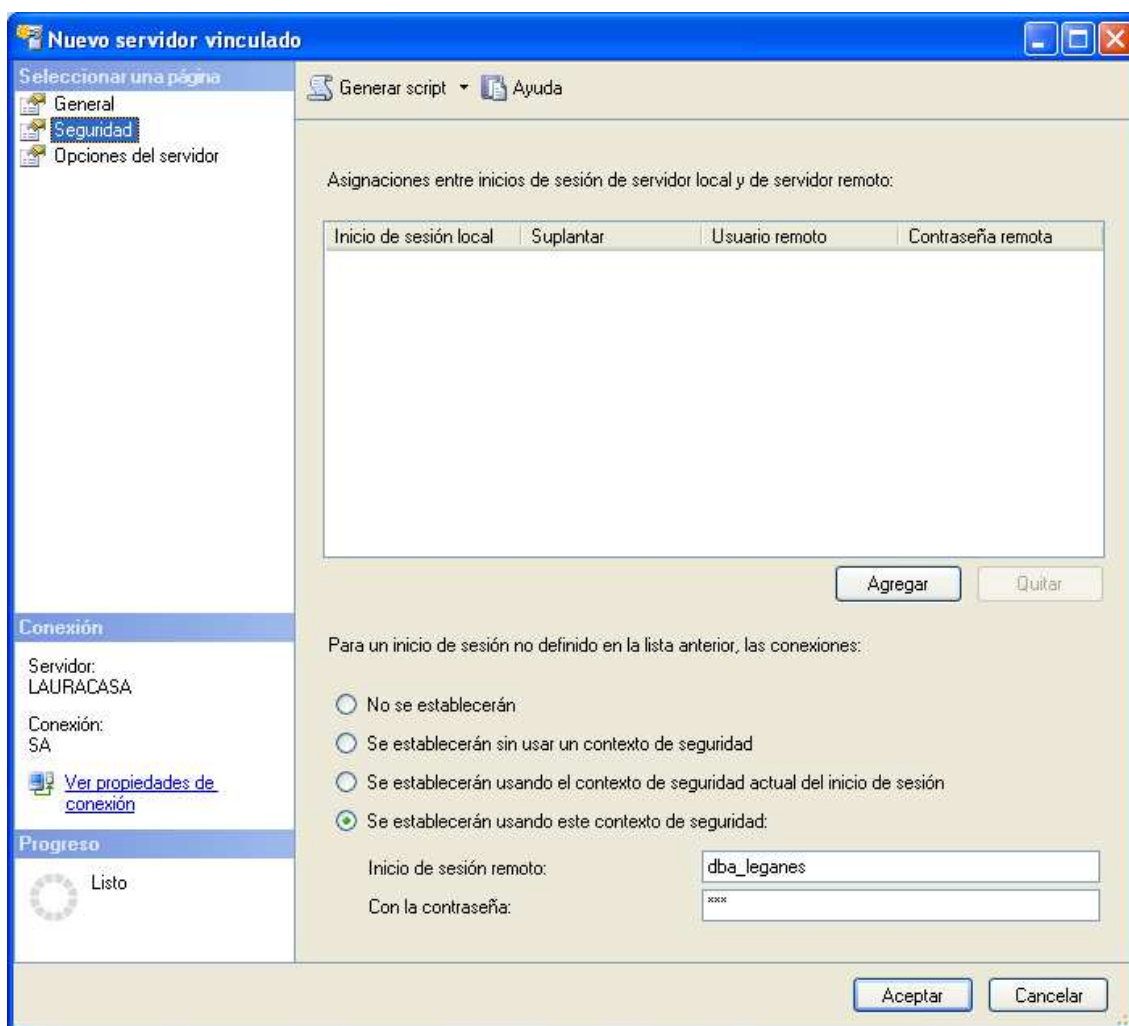
**PROVEEDOR:** Seleccionamos “Microsoft OLE DB Provider for Oracle”.

**NOMBRE DEL PRODUCTO:** Oracle

**ORIGEN DE DATOS:** XE

**CADENA DE PROVEEDOR:** MSDAORA

Y en la opción “Seguridad” nos aparece una pantalla similar a la Ilustración 31.



**Ilustración 31 Nuevo Servidor Vinculado - seguridad.**

Se selecciona la opción “Se establecerán usando este contexto de seguridad” que nos habilita los cuadros de texto de “Inicio de sesión remoto” y “Con la contraseña” donde introduciremos las credenciales de un usuario con permiso de conexión existente en la BD Oracle objetivo de la conexión. En este caso añadimos el usuario “dba\_leganes”, con permisos de administrador, en la BD de la sede de Leganés.

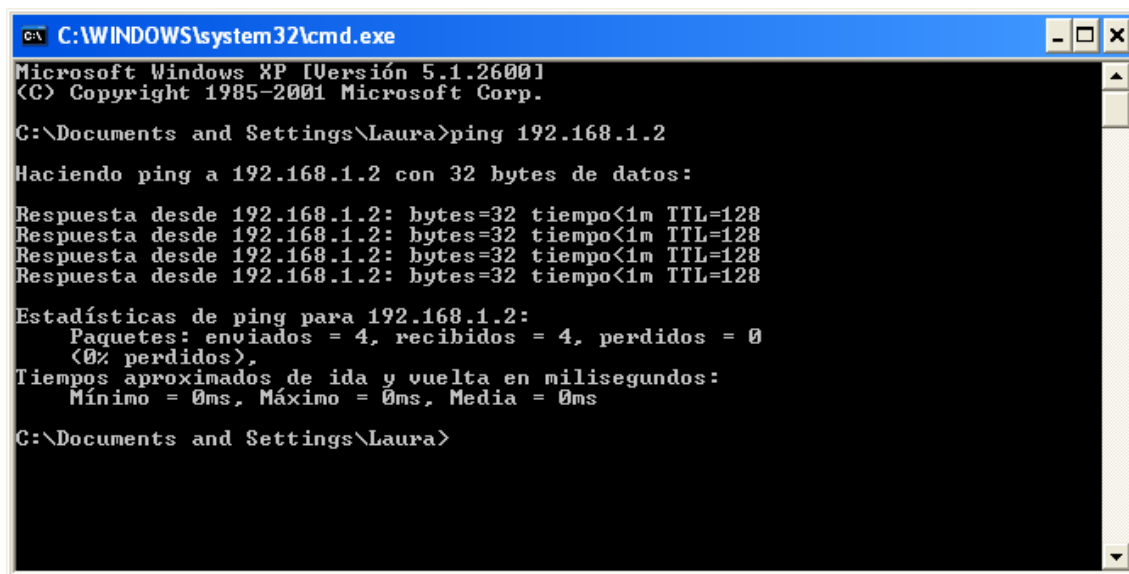
## 6 Experimentación y Resultados

A continuación se van a exponer una serie de pruebas realizadas para llevar a cabo una implementación del sistema

### 6.1 Prueba de conectividad de red entre las sedes

En primer lugar se realiza una prueba para verificar que las sedes están conectadas en red. Veamos un ejemplo de cómo realizar esta prueba desde la sede de Leganés a Getafe

en la Ilustración 32. Ejecutamos el comando ping<sup>3</sup> desde la consola de comandos de Windows



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Laura>ping 192.168.1.2

Haciendo ping a 192.168.1.2 con 32 bytes de datos:

Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.1.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Documents and Settings\Laura>
```

Ilustración 32 Prueba de conexión de red desde Sede de Leganés a sede de Getafe

Es necesario realizar la misma prueba en la dirección opuesta, es decir desde la sede de Getafe a Leganés ejecutando el mismo comando.

## 6.2 Prueba del servidor vinculado

Probamos el servidor vinculado creado en el apartado 5.3.2 para conectar la sede de Getafe con la de Leganés, realizando la siguiente consulta:

```
select * from
ENLACE_GETAFE_LEGANES..dba_leganes.ASIGNATURAS_LEGANES
```

Que será la forma en la que podrán realizar consultas desde la sede de Getafe a la de Leganés, donde:

**ENLACE\_GETAFE\_LEGANES:** nombre del servidor vinculado

**dba\_leganes:** nombre del usuario propietario de las tablas.

**ASIGNATURAS\_LEGANES:** es la tabla remota a la que queremos acceder

Cada instancia de una base de datos Oracle sólo tiene un catálogo con un nombre vacío. Se debe hacer referencia a las tablas de un servidor vinculado Oracle usando un nombre de cuatro partes con el formato

OracleLinkedServerName..OwnerUserName.TableName.

<sup>3</sup> El comando ping es un gestor de transporte de paquetes y es comúnmente utilizado para probar la conexión entre diferentes nodos de una red.

Por ejemplo, esta instrucción `SELECT` hace referencia a la tabla `ASIGNATURAS_LEGANES` que es propiedad del usuario Oracle `dba_leganes` en el servidor asignado por el servidor vinculado `OrclDB`.

Se usarán las siguientes reglas cuando se haga referencia a las tablas de un servidor vinculado Oracle:

Si los nombres de tabla y de columna fueron creados en Oracle sin identificadores entre comillas, necesitaremos usar nombres en mayúsculas.

Si los nombres de tabla y de columna fueron creados en Oracle con identificadores entre comillas, se usarán exactamente los mismos nombres que se usaron cuando se crearon en Oracle.

### 6.3 Prueba de enlaces públicos

Probamos el servidor enlace público creado en el apartado 5.3.1 para conectar la sede de Leganés con la de Getafe, realizando la siguiente consulta:

```
Select * from dual@enlace_leganes_getafe
```

Que nos devuelve lo siguiente:

DUMMY
X
1 filas devueltas en 0,46 segundos

La configuración se ha realizado con éxito y desde este momento, siempre que el servidor de Getafe esté disponible, se podrá acceder a sus objetos designándolos como `<nombre_objeto>@ENLACE_LEGANES_GETAFE`.

### 6.4 Prueba de los disparadores que garantizan la transparencia de la fragmentación vertical

Se crea el siguiente Trigger para hacer la fragmentación vertical de la tabla `PROFESORES_AUX`. Lo que se busca es lograr que para el usuario las inserciones en la tabla profesores sean transparentes, es decir no queremos que el usuario haga un insert en la tabla `NOMINAS_PROFESORES` e `INFO_PROFESORES` de cada sede, para el usuario debe ser como si solo existiera una tabla, la tabla `PROFESORES_AUX`. Este tipo de objeto, el Trigger, es una “función” que se va a ejecutar antes o después de una transacción, en este caso después de una transacción de tipo insert en la tabla `PROFESORES_AUX`. Esta tabla tiene el sinónimo `PROFESORES`:



El Trigger trata de hacer un insert en la tabla de la sede Leganés INFO\_PROFESORES\_LEGANES y otro en la tabla NOMINAS\_PROFESORES\_GETAFE con los datos asignados en una tupla recién añadida a la tabla PROFESORES\_AUX. Compila sin problema y lo probamos con una inserción de prueba sobre la tabla PROFESORES\_AUX utilizando el sinónimo PROFESORES que se ha creado sobre esta tabla.

```
insert into PROFESORES(NOMBRE, EMAIL, DESPACHO, direccion,
telefono, categoria, id_pluses_hijo)
values ('nombre prueba', 'email prueba', 'desp', 'dir
prueba', '123456', 'L-1', 'L-1')
```

Nos muestra el siguiente error:

```
Error que empieza en la línea 15 del comando:
insert into PROFESORES (NOMBRE, EMAIL, DESPACHO,direccion,
telefono, categoria, id_pluses_hijo)
values ('nombre prueba', 'email prueba', 'desp','dir prueba',
'123456', 'L-1', 'L-1')
Informe de error:
Error SQL: ORA-02047: no se puede unir la transacción
distribuida en curso
ORA-06512: en "DBA_LEGANES.FRAGMENTA_PROFESORES_AUX", línea 5
ORA-04088: error durante la ejecución del disparador
'DBA_LEGANES.FRAGMENTA_PROFESORES_AUX'
02047. 00000 - "cannot join the distributed transaction in
progress"
*Cause:      Either a transaction is in progress against a remote
database
              that does not fully support two phase commit, and an
update
              is attempted on another database, or updates are
pending and
              and an attempt is made to update a different database
that
              does not fully support two phase commit.
*Action:     complete the current transaction and then resubmit
the
              update request.
```

El error “ORA-02047: cannot join the distributed transaction in progress” nos indica que no puede continuar la transacción porque antes de continuar con otra transacción necesita hacer commit de la transacción distribuida.

Para evitar el error vamos a añadir al Trigger la directiva **PRAGMA AUTONOMOUS\_TRANSACTION**. Con esta directiva marcamos el subprograma para que se comporte como transacción diferente a la del proceso principal, llevando el control de **COMMIT** o **ROLLBACK** independiente. El Trigger quedará de la siguiente manera:

```

CREATE OR REPLACE TRIGGER fragmenta_PROFESORES_AUX
AFTER INSERT
  ON dba_leganes.PROFESORES_AUX
  REFERENCING NEW AS NEW
  FOR EACH ROW
  DECLARE PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  insert into INFO_PROFESORES_LEGANES(ID_INFO_PROFESOR,NOMBRE,
EMAIL,  DESPACHO)
  values (:NEW.ID_PROFESOR_AUX, :NEW.NOMBRE, :NEW.EMAIL,
:NEW.DESPACHO);
  COMMIT;
  insert into
NOMINAS_PROFESORES_GETAFE@ENLACE_LEGANES_GETAFE(ID_NOMINAS_PROFE
SOR, direccion, telefono, categoria, id_pluses_hijo)
  values (:NEW.ID_PROFESOR_AUX, :NEW.DIRECCION,
:NEW.TELEFONO, :NEW.CATEGORIA, :NEW.ID_PLUSES_HIJO);
  COMMIT;
END fragmenta_PROFESORES_AUX;
/

```

Ahora volvemos a intentar hacer el insert en la tabla “PROFESORES” y éste es realizado como se esperaba, es decir al realizar un insert sobre la tabla PROFESORES se ejecutan dos insert uno para cada uno de los fragmentos verticales en los que está dividida esta tabla.

## 6.5 Consulta y recuperación de los datos de una tabla fragmentada horizontalmente

Se realiza una consulta a la tabla TITULACIONES, con la intención de recuperar los datos que componen esta tabla en todas las sedes:

ID_TITULACION	NOMBRE	CREDITOS	NOTA_MINIMA	CAMPUS
G-5	PERIODISMO	200	0	GETAFE
L-1	Ingenieria Tecnica en Informatica de Gestion	120	5,5	Leganes
L-2	Ingenieria en Informatica	200	6,5	Leganes
L-3	Grado en Informatica	150	6,5	Leganes

## 6.6 Consulta y recuperación de los datos de una tabla fragmentada vertical y horizontalmente

Se realiza una consulta a la tabla PROFESORES, con la intención de recuperar los datos que componen esta tabla en todas las sedes. Esta tabla está fragmentada vertical y horizontalmente:

```
Select * from v_profesores
```

Esta consulta devuelve los datos de todos los profesores de la universidad, tanto referentes a las nominas como a la información de sus datos profesionales.

ID_INFO_PROFESOR	NOMBRE	EMAIL	DESPACHO	DIRECCION	CATEGORIA	TELEFONO	ID_PLUSES_HIJO
L-21	nombre prueba	email prueba	desp	dir prueba	L-1	123456	L-1

Aquí faltaría añadir un comentario sobre si el resultado de cada una de las pruebas es o no satisfactorio conforme a los objetivos esperados. Valdría por ejemplo insertar una tablita con la lista de pruebas realizadas y el veredicto para cada una de ellas (Satisfactorio o No satisfactorio) explicando brevemente si se cumplen los objetivos cuantitativa y/o cualitativamente, es decir, los resultados (Algo breve, pero que no debe faltar)

## 7 Conclusiones

Una vez desarrollado este trabajo de fin de carrera las principales conclusiones que se han obtenido del mismo han sido las que se citan a continuación:

- se han estudiado las posibilidades que dos de los SGBD más importantes del mercado (Oracle y SQL Server) ofrecen a la hora de implementar sistemas de bases de datos distribuidas. Ambos SGBD tienen herramientas para conectar con otros SGBD, como son dblink (Servicios Heterogéneos de Oracle) y Servidor Vinculado (MS SQL Server). Tras la finalización del caso práctico se han adquirido las habilidades necesarias para utilizar estas herramientas y sus respectivos objetos.
- en la resolución del caso práctico desarrollado en este documento, se han tenido que crear varios objetos (trigger, sinónimos, vistas) por cada tabla para intentar llegar a “simular” una transparencia de red, de fragmentación, de asignación y de réplica sobre los datos. En la teoría los conceptos y las necesidades son bastante claros pero a la hora de implementar este tipo de sistemas el DBA debe hacer un esfuerzo importante. Resumiendo, a pesar de ser un caso pequeño, de menos de 10 tablas, la tarea de administración a realizar ha sido trabajosa, quedando expuesta la carencia de los actuales SGBD y pudiendo aprender o desarrollar los “trucos” para simular un sistema centralizado.
- en cuanto al interés por los conocimientos teóricos y prácticos que motivaron este proyecto podemos concluir que se han adquirido con éxito, habiéndose satisfecho los objetivos planteados y completándose las pruebas de validación de la implementación final, producto de todo el trabajo de estudio y desarrollo previo.

## 8 Líneas Futuras

Como se ha mencionado en el apartado 7, la parte más complicada de los sistemas distribuidos es proveer de una transparencia de distribución total al usuario y, además, en los sistemas distribuidos heterogéneos, de una transparencia en cuanto al lenguaje. Por todo lo mencionado, una de las líneas futuras a llevar a cabo podría ser el desarrollar una implementación con los objetos necesarios para simular la transparencia de todo el sistema distribuido o una aplicación capaz de gestionar las conexiones y la transparencia de una forma más sencilla que la actual.

Otra interesante línea futura sería el realizar la conexión entre las diferentes BD con varios métodos o herramientas de conexión y evaluar la eficiencia y rendimiento de éstas. Entre las posibles herramientas de conexión que se podrían probar están:

- **Attunity**, que proporciona integración de datos en tiempo real y el software de captura de eventos que permite el acceso y la entrega de datos a través de fuentes heterogéneas sin fisuras. Un pionero en el acceso a datos federados y cambio de datos de captura en tiempo real.
- Para la conexión con SQL Server existe la herramienta **SQL Server Integration Services (SSIS)**. SSIS es un producto que viene incluido con la licencia de Microsoft SQL Server y que es la propuesta de Microsoft para lo que se llama Extracción, Transformación y Carga de datos (también conocidos como procesos ETL).  
SSIS es un producto en el que se desarrolla de forma muy visual, pinchando y arrastrando componentes desde la barra de herramientas hasta los lugares de diseño, cada uno de estos componentes admitirá parametrización para adaptarlo a diferentes necesidades [0].
- **Oracle Transparent Gateways**, esta tecnología está compuesta de dos partes: un componente que posee la tecnología genérica para conectar a sistemas no Oracle, el cual es comúnmente conocido como servicios heterogéneos, y un componente que es específico de los sistemas no Oracle que se conectan a través de Gateway. Los servicios heterogéneos, en conjunto con el agente de Transparent Gateway, capacita de transparencia el acceso a sistemas no Oracle desde un entorno Oracle [10].

## 9 Lista de Acrónimos

<b>Acrónimo</b>	<b>Significado</b>
<b>ANSI-SPAR</b>	American National Standard Institute - Standards Planning and Requirements Committee
<b>API</b>	Application Programming Interfaces (Interfaz de Programación de Aplicaciones).
<b>B2F</b>	Bloqueo de dos fases
<b>BD</b>	Base Datos.
<b>BDD</b>	Bases Datos Distribuida.
<b>CW</b>	Cautions waiting
<b>DBA</b>	Database Administrator (Administrador de Base de Datos)
<b>DBMS</b>	DataBase management system.
<b>DD</b>	Diccionario de Ddatos.
<b>ECG</b>	Esquema Conceptual Global
<b>ECL</b>	Esquema Conceptual Local
<b>ELG</b>	Esquema lógico global.
<b>ELL</b>	Esquema lógico local.
<b>ETL</b>	Extracción, Transformación y Carga de datos
<b>GTM</b>	Administrador de transacciones globales
<b>LAN</b>	Local Area Network (Red de área local)
<b>LMD</b>	Lenguaje de manipulación de datos
<b>LTM</b>	gestores de transacciones locales
<b>MT</b>	Marca de tiempo
<b>MT</b>	marcas de tiempo
<b>NW</b>	No waiting
<b>ODAC</b>	Oracle Data Access Components
<b>OLE DB</b>	Object Linking and Embedding for Databases
<b>OMT</b>	Ordenamiento por marca de tiempo
<b>OMT</b>	ordenamiento por marcas de tiempo
<b>RL</b>	Read lock. Candado de lectura
<b>SBD</b>	Sistema de base de datos
<b>SBDD</b>	Sistemas de bases de datos distribuidas.
<b>SGBD</b>	Sistema gestor de base de datos.
<b>SGBDD</b>	Sistema gestor de base de datos distribuida.
<b>SMBD</b>	Sistema de multibase de datos.
<b>SPJ</b>	Select-project-join
<b>SSIS</b>	SQL Server Integration services
<b>STG</b>	Subtransacción global
<b>TG</b>	Transacción global
<b>WL</b>	Write lock. Candado de escritura

## 10 Bibliografía

### 10.1 Referencias bibliográficas

1. ELMASRI, Navathe, *Fundamentos de Sistemas de Bases de Datos*. Shamkant B. Navathe; traducción, Verónica Canivell Castillo [et al.]; revisión técnica, Alfredo Goñi Sarriuren [et al.] 2001. 962p. ISBN: 84-782-9051-6.
2. CUADRA, María Dolores, et al. *Desarrollo de Bases de datos: casos prácticos desde el análisis a la implementación*. Mención de responsabilidad secundaria (traductor; prologuista; ilustrador; coordinador; etc.)\*. N° de edición. Lugar de edición Paracuellos de Jarama (Madrid): Ra-Ma, 2007. 569 p. ISBN: 9788478978359.
3. DYE, Charles. *Oracle Distributed Systems*. O'Reilly Media, Inc, 1999. 522p. ISBN: 978-1-56592-432-1.
4. EIBE GARCÍA, Santiago. *Bases de datos distribuidas*. ED: Fundación General de la U.P.M, D.L. 2006. Descripción Física: 112 p. ISBN: 849673711X
5. ROB, Peter. *Sistemas de Bases de Datos: diseño, implementación y administración*. Coronel, Carlos; traducción, Rodolfo Navarro Salas; 5° Edición, México D.F. [etc.]: Thomson, cop. 2004. Descripción Física: 838 p. ISBN: 9706862862
6. ÖZSU, Tamer. *Principles of distributed database system*. Segunda edición, Prentice-Hall International Edition. 664 p. Encuadernación: Tapa dura. ISBN: 9780136597070
7. CARDENAS, A.F. *Heterogeneous Distributed Database management: The HD-DBMS*. Este artículo aparece en: Proceedings of the IEEE. 2005. Páginas: 588 - 600. ISSN: 0018-9219.
8. SHETH, Amit P. *Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases*. James A. Larson. ACM Computing volumen 22, Edición 3 (Septiembre 1990). Páginas: 183 – 236. ISSN: 0360-0300.
9. HSIAO, David K. *Heterogeneous Databases: Proliferations, Issues, and Solutions*. Este artículo aparece en Transactions on knowledge and data engineering, VOL. I, NO. I. 1989. Bellow, IEEE, and magdin.kamel, member, IEEE
10. MULLIS, Craig S. *Database Administration: The Complete Guide to Practices and Procedures*. Addison-Wesley Professional, 2002. 736p. ISBN: 978-0-201-74129-2

## ***10.2 Referencias en Internet***

1. Redes de computadores (Último acceso 03/11/2009)  
<http://es.wikipedia.org>
2. Arquitectura de los sistemas de bases de datos (Último acceso 03/11/2009)  
<http://www.monografias.com/>
3. Sistema de Gestión de Base de Datos (Último acceso 17/08/2010)  
<http://es.wikipedia.org>
4. Base de Datos Distribuida (Último acceso 17/08/2010)  
<http://tecnomaestros.awardspace.com>
5. Algebra relacional (Último acceso 23/08/2010)  
<http://es.wikipedia.org>
6. Servidor Vinculado y Analysis Services (Último acceso 30/08/2010)  
<http://msdn.microsoft.com>
7. Lenguajes Comerciales Relacionales en Bases de Datos (Último acceso 13/09/2010)  
<http://www.um.es/docencia/barzana/IAGP/Iagp9.html>
8. Propiedades del servidor vinculado (página Opciones del proveedor) (Último acceso 27/09/2010)  
<http://msdn.microsoft.com>
9. SQL Server Integration Services (SSIS) (Último acceso 02/06/2010)  
<http://technet.microsoft.com>
10. Oracle Transparent Gateway for Microsoft SQL Server Administrator's Guide 10g Release 1 (10.1) for Microsoft Windows (Último acceso 06/27/10/2010)  
<http://download.oracle.com/docs>

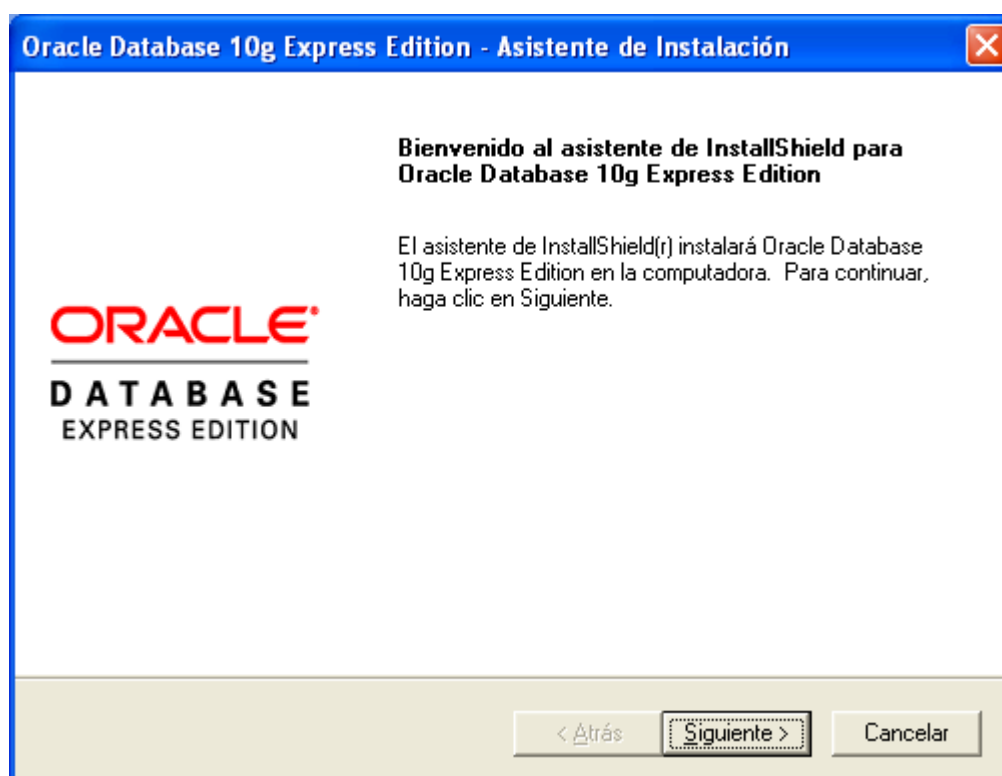
## 11 Anexos

### 11.1 Instalación de Oracle 10G Express Edition

En este ANEXO se procede a describir la instalación paso a paso de Oracle 10g Express Edition

En primer lugar, se descarga el fichero *OracleXE.exe* de la Web de Oracle:

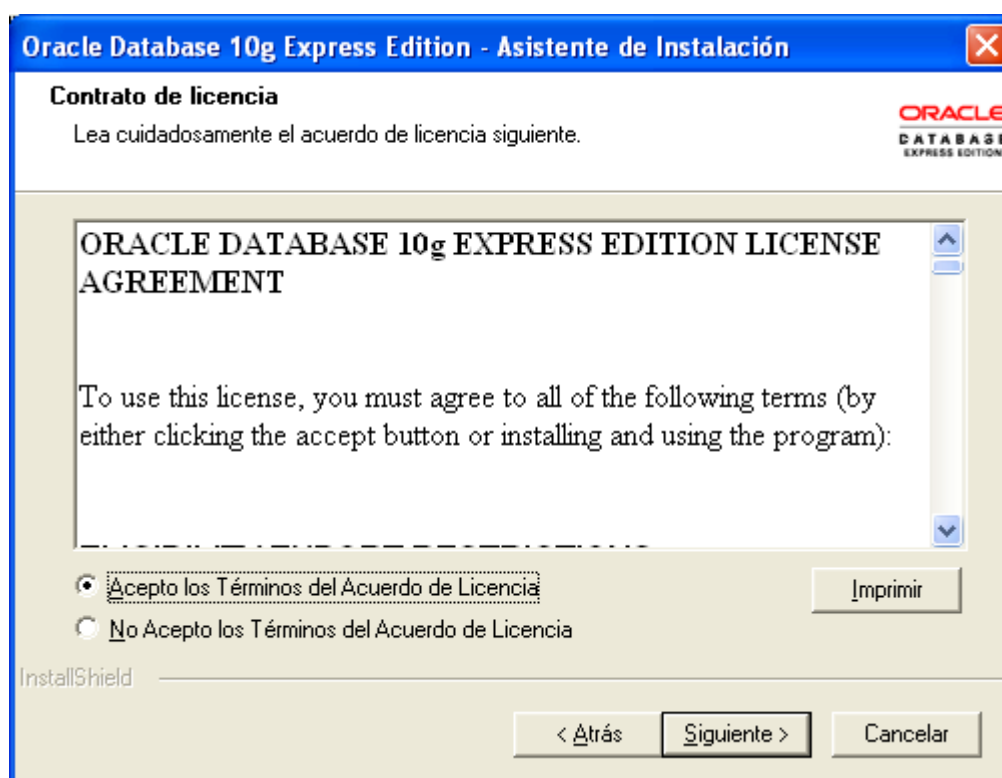
Ejecutamos el fichero descargado (ver Ilustración 33), pulsaremos *Siguiente* para iniciar la instalación:



**Ilustración 33 Oracle Database 10g Express Edition – Asistente de Instalación 1/6**

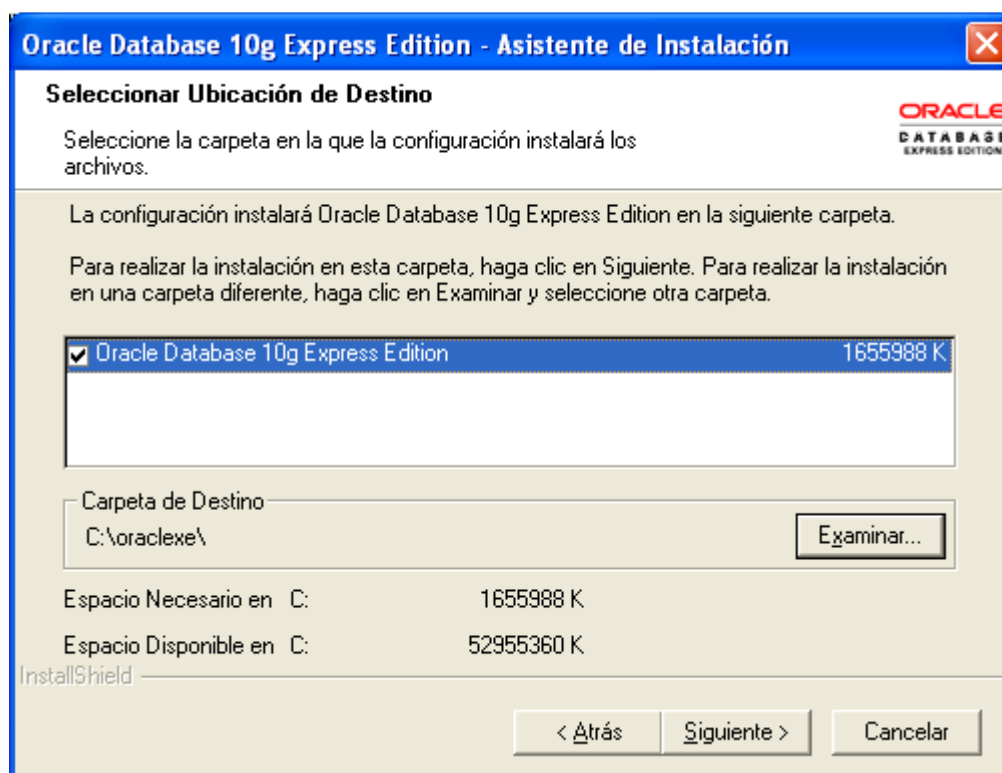
En la siguiente pantalla, la Ilustración 34 nos solicitará que aceptemos el contrato de licencia. Aceptamos y damos a “Siguiente”.





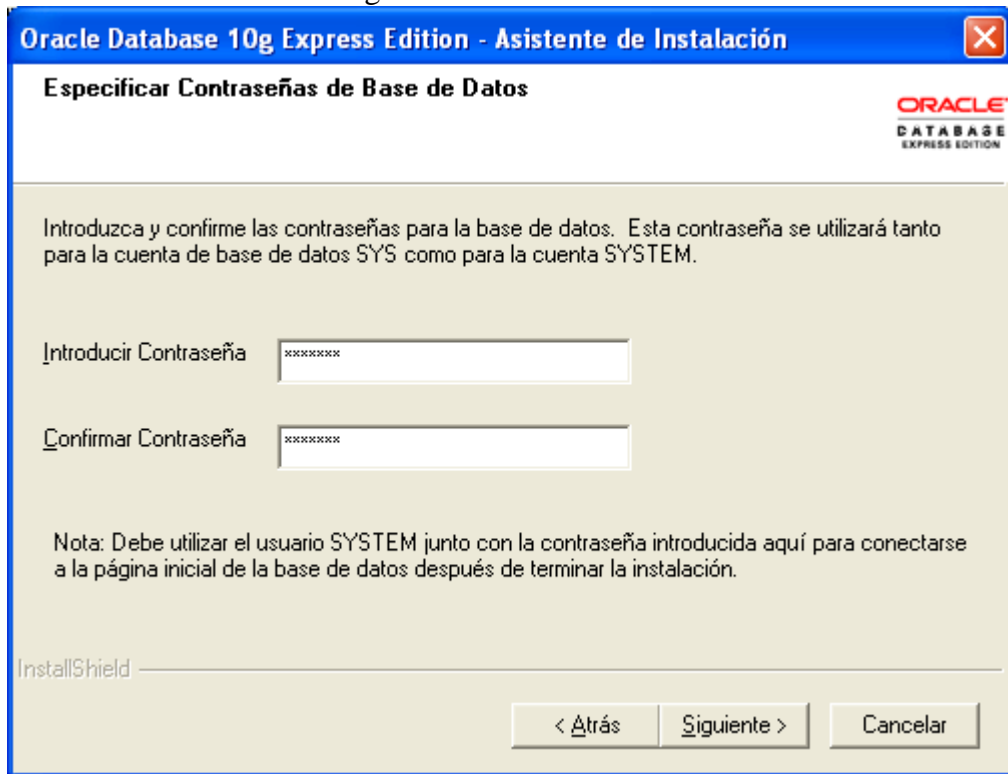
**Ilustración 34 Oracle Database 10g Express Edition – Asistente de Instalación 2/6**

En la Ilustración 35 nos solicita una ruta donde realizar la instalación y nos indica el espacio mínimo requerido para realizarla. Dejaremos la que viene por defecto “C:\oraclexe” y pulsamos “Siguiete”.



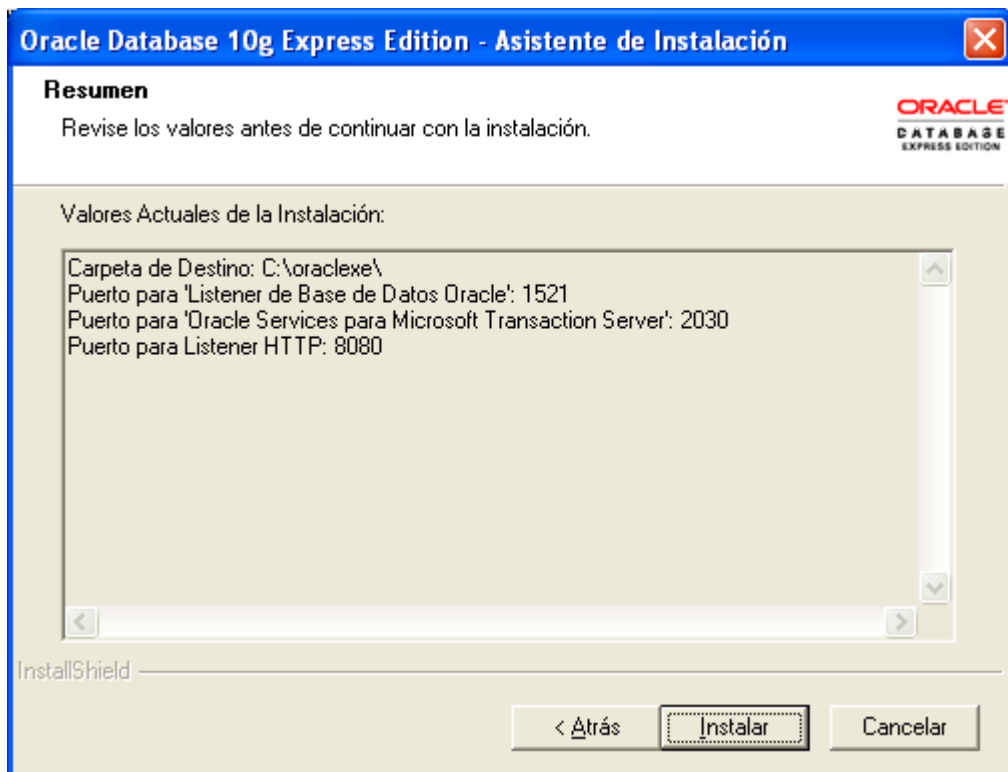
**Ilustración 35 Oracle Database 10g Express Edition – Asistente de Instalación 3/6**

En la Ilustración 36 nos solicita una contraseña para conectarnos como usuario “SYS” o “SYSTEM”. Es necesario introducir la misma contraseña en los dos cuadros de texto. Pulsamos “Siguiente”.



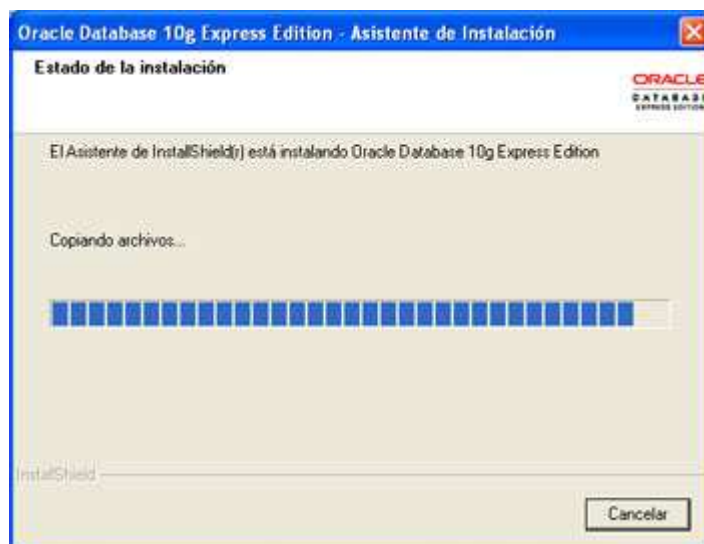
**Ilustración 36 Oracle Database 10g Express Edition – Asistente de Instalación 4/6**

A continuación en la Ilustración 37 aparece una ventana con las opciones de instalación elegidas. Pulsamos “Instalar” para iniciar el proceso:



**Ilustración 37 Oracle Database 10g Express Edition – Asistente de Instalación 5/5**

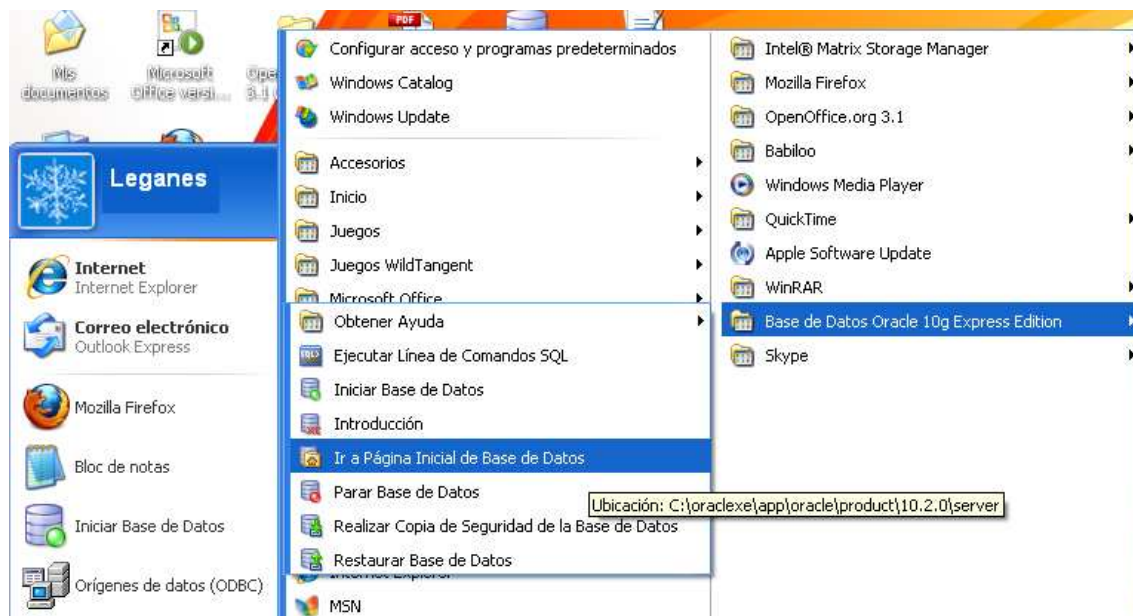
En la Ilustración 38 se observa el proceso de copia de ficheros y el proceso de configuración automática de la base de datos. Por defecto, el instalador de Oracle 10g Express Edition, crea y configura una base de datos.



**Ilustración 38 Oracle Database 10g Express Edition – Asistente de Instalación 6/6**

Una vez que termina la instalación nuestro SGBD está preparado para ser utilizado. Accedemos a la página de configuración de la base de datos, como se muestra en la Ilustración 39, de la siguiente manera:

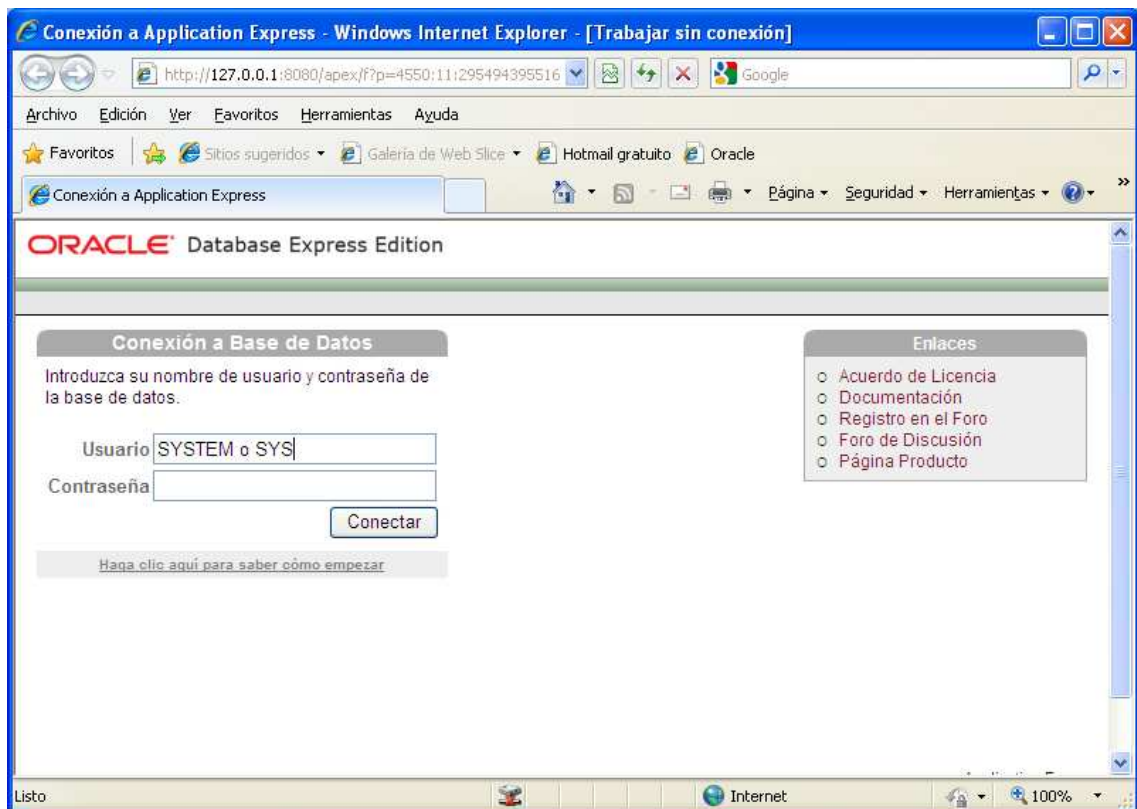
**Inicio >> Programas >> Base de Datos Oracle 10g Express Edition >> Ir a Página Inicial de Base de Datos**



**Ilustración 39 Cómo acceder a la página de configuración y administración de Oracle**

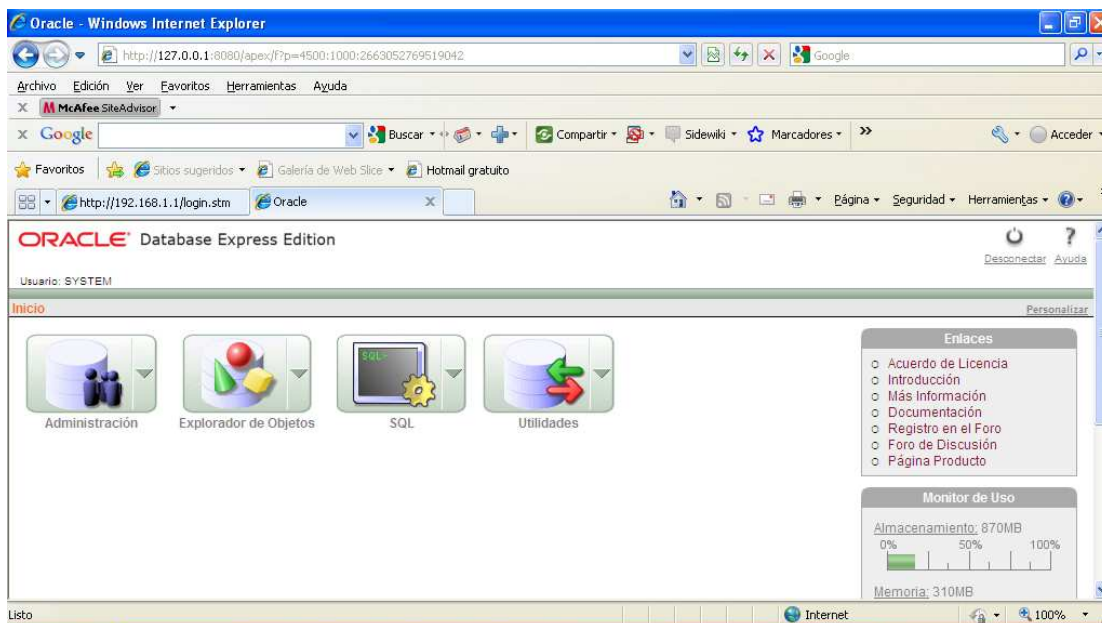
Nos aparecerá la página Web para administrar Oracle 10g Express Edition, accesible introduciendo en el explorador de Internet: <http://127.0.0.1:8080/apex>. Aquí es donde deberemos introducir la clave que especificamos durante la instalación para el

usuario “SYS” o “SYSTEM” (Véase la Ilustración 40). Más adelante podremos crear otros usuarios:



**Ilustración 40** Validación en el acceso a la página de configuración y administración de Oracle.

Nos aparece una ventana de administración (limitada con respecto a las versiones completas) con varias opciones, como podemos ver en la Ilustración 41: **Administración** (para configurar las opciones de almacenamiento, memoria, usuarios y monitorización), **Explorador de objetos** (para visualizar, modificar y crear tablas, vistas, índices, funciones, triggers, procedimientos, paquetes, secuencias, etc.), **SQL** (para ejecutar consultas SQL, scripts, etc.), **Utilidades** (exportación, importación, papelera de reciclaje, informes, generación de sentencias DDL, etc.):



**Ilustración 41 Ventana de administración de Oracle 10g Express Edition**

El programa de instalación de Oracle 10g Express Edition habrá creado los servicios que se muestran en la Tabla 15:

Nombre	Ubicación	Inicio
OracleJobSchedulerXE	c:/oraclexe/app/oracle/product/10.2.0/server/Bin/ex tjob.exe XE	Deshabilitad o
OracleMTSRecoveryService	C:/oraclexe/app/oracle/product/10.2.0/server/BIN/o mtsreco.exe "OracleMTSRecoveryService"	Manual
OracleServiceXE	c:/oraclexe/app/oracle/product/10.2.0/server/bin/O RACLE.EXE XE	Automático
OracleXEClrAgent	C:/oraclexe/app/oracle/product/10.2.0/server/bin/O raClrAgnt.exe	Manual
OracleXETNSListener	C:/oraclexe/app/oracle/product/10.2.0/server/BIN/t nlsnr.exe	Automático

**Tabla 15 Servicios de Oracle 10g Express Edition**

Los accesos directos que crea la aplicación se muestran en la Tabla 16:

Acceso directo	Descripción
Introducción	Muestra una ventana de ayuda en formato HTML
Ir a Página Inicial de Base de Datos	Muestra la página de configuración y administración de Oracle
Parar Base de Datos	Detiene la base de datos
Realizar Copia de Seguridad de la Base de Datos	Copia de seguridad de la base de datos

Restaurar Base de Datos	Restaura una copia de seguridad de la base de datos
Obtener Ayuda	Ayuda
Ejecutar Línea de Comandos SQL	Permite ejecutar comandos SQL desde una ventana de MS-DOS
Iniciar Base de Datos	Inicia la base de datos previamente detenida

**Tabla 16 Accesos directos de Oracle 10g Express Edition**

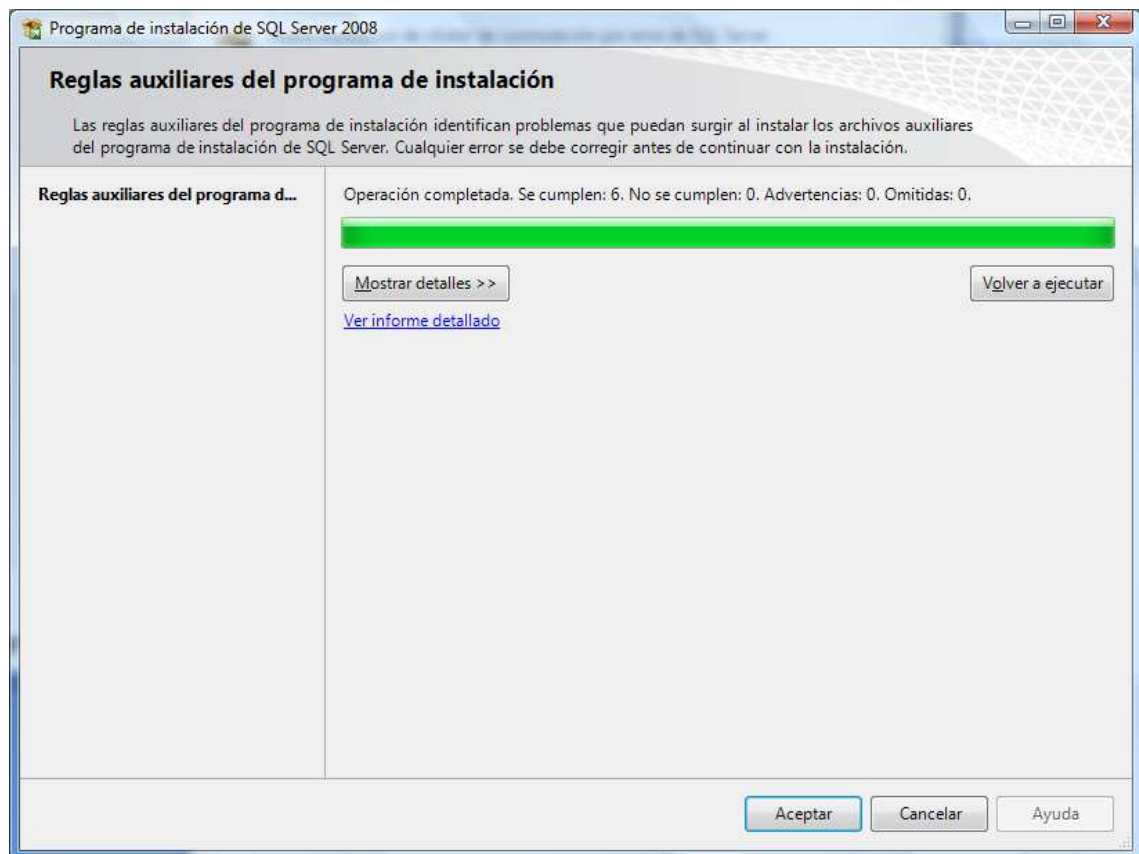
## ***11.2 Instalación de Microsoft SQL Server 2008***

Para proceder a la instalación de SQL Server 2008 se realizarán los siguientes pasos:  
En la primera pantalla que nos aparece se selecciona la opción “Nueva instalación independiente de SQL Server o agregar características a una instalación existente” y nos lleva a la pantalla que muestra la Ilustración 42



**Ilustración 42 Centro de Instalación de SQL Server**

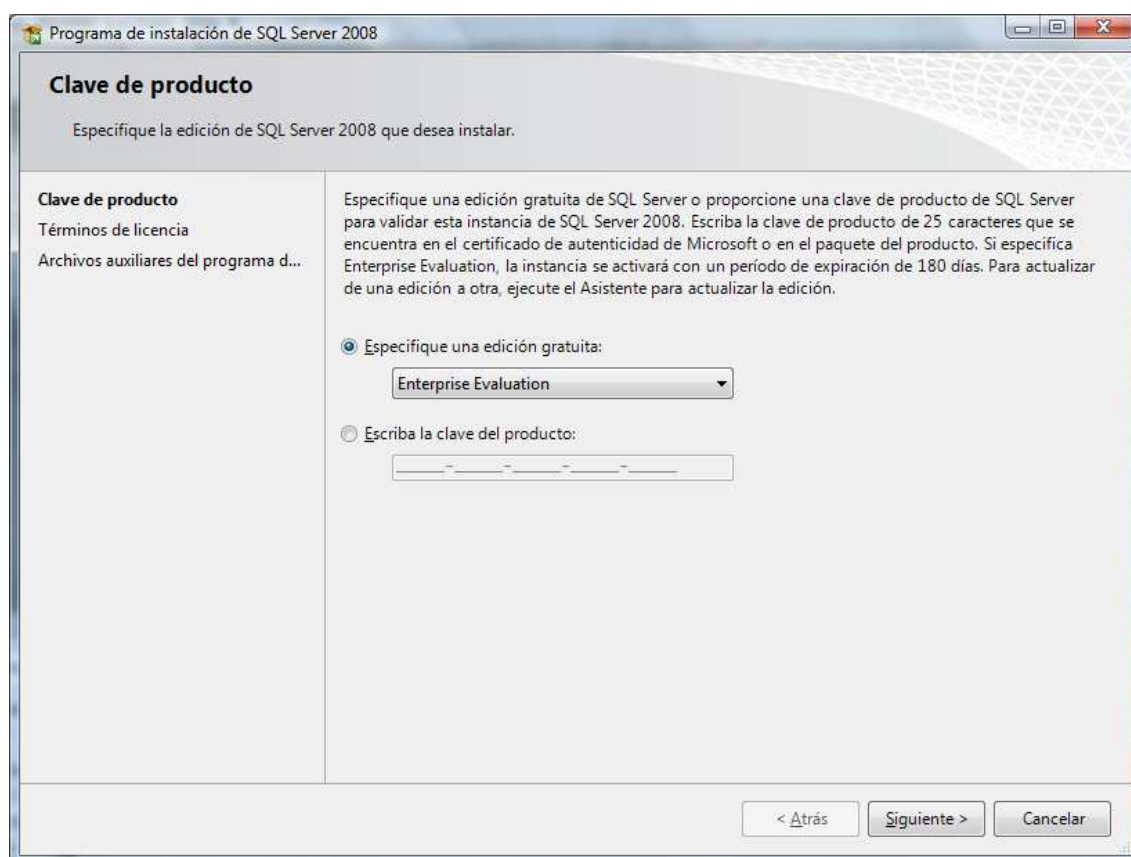
Se realizan unas comprobaciones, todo ha ido correctamente y pulsamos aceptar. Esta opción conduce a la pantalla de la Ilustración 43.



**Ilustración 43 Programa de instalación de SQL Server 2008 1/21**

La versión de que disponemos es de prueba y por ello en la pantalla de la Ilustración 44 dejamos la opción que viene por defecto “Enterprise Evaluation” y pulsamos “Siguiente”.

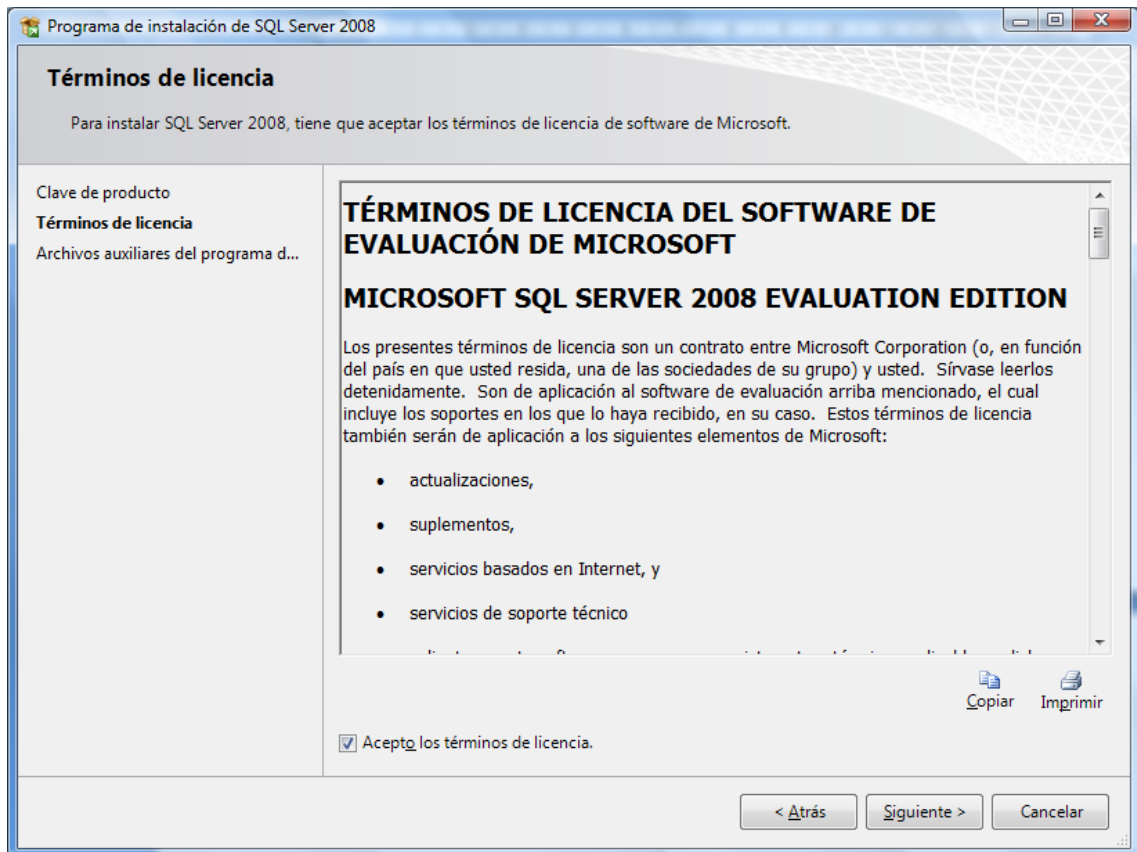




**Ilustración 44 Programa de instalación de SQL Server 2008 2/21**

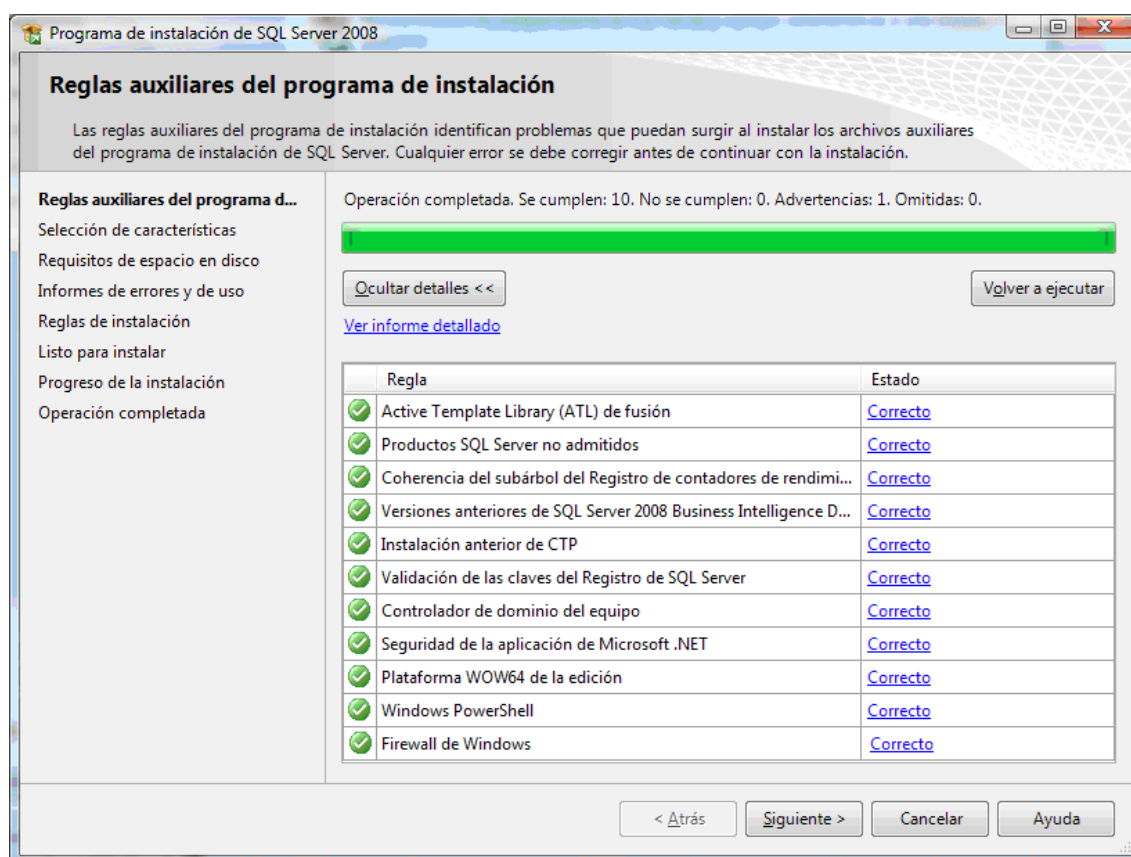
En la pantalla de Ilustración 45 nos solicitará que aceptemos los términos de acuerdo de licencia.





**Ilustración 45 Programa de instalación de SQL Server 2008 3/21**

Nos lleva a la siguiente pantalla que nos muestra donde pulsamos “Instalar”.En la Ilustración 46 muestra una pantalla en la que se realizan una serie de comprobaciones y no nos aparece ningun error, todo es correcto para continuar con la instalación.



**Ilustración 46 Programa de instalación de SQL Server 2008 4/21**

En esta pantalla pulsaremos “Seleccionar Todo”, dejaremos la ruta del directorio de características compartidas que viene por defecto y pulsaremos

Siguiente”

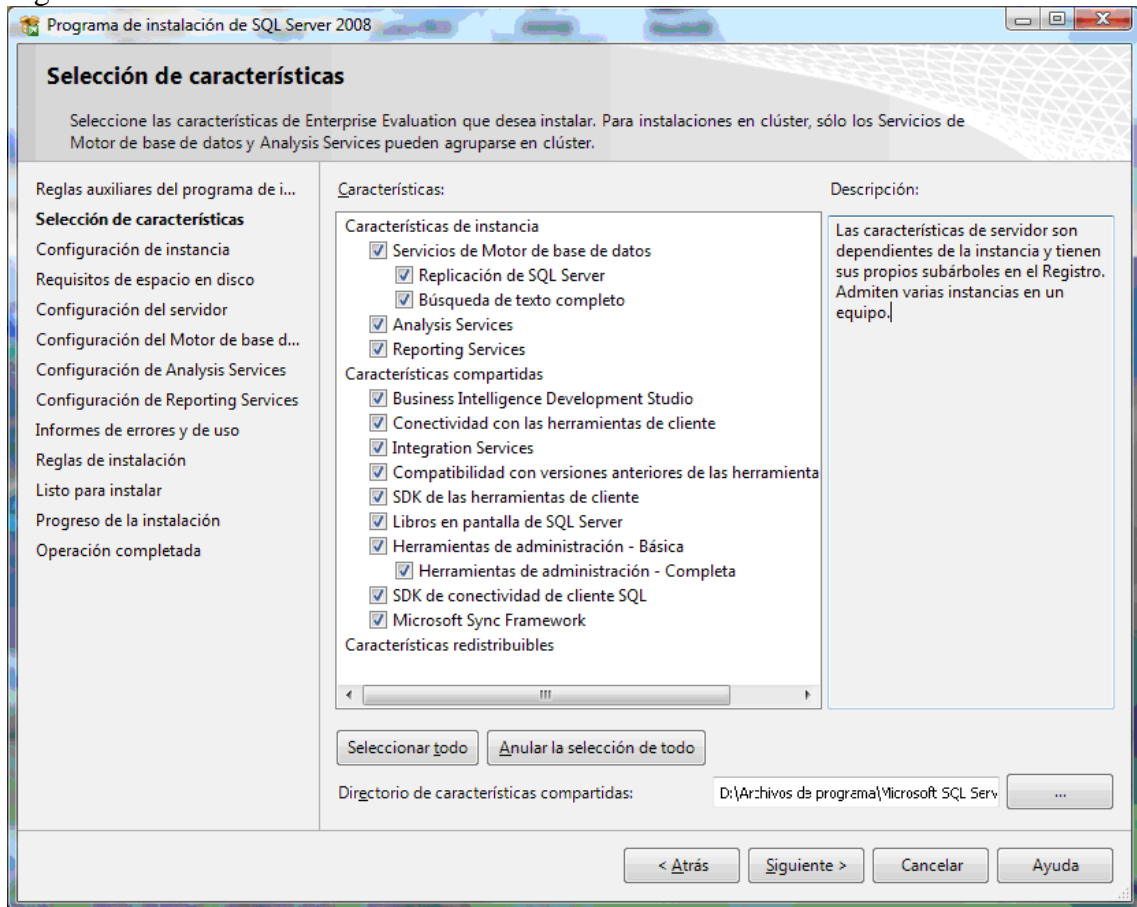
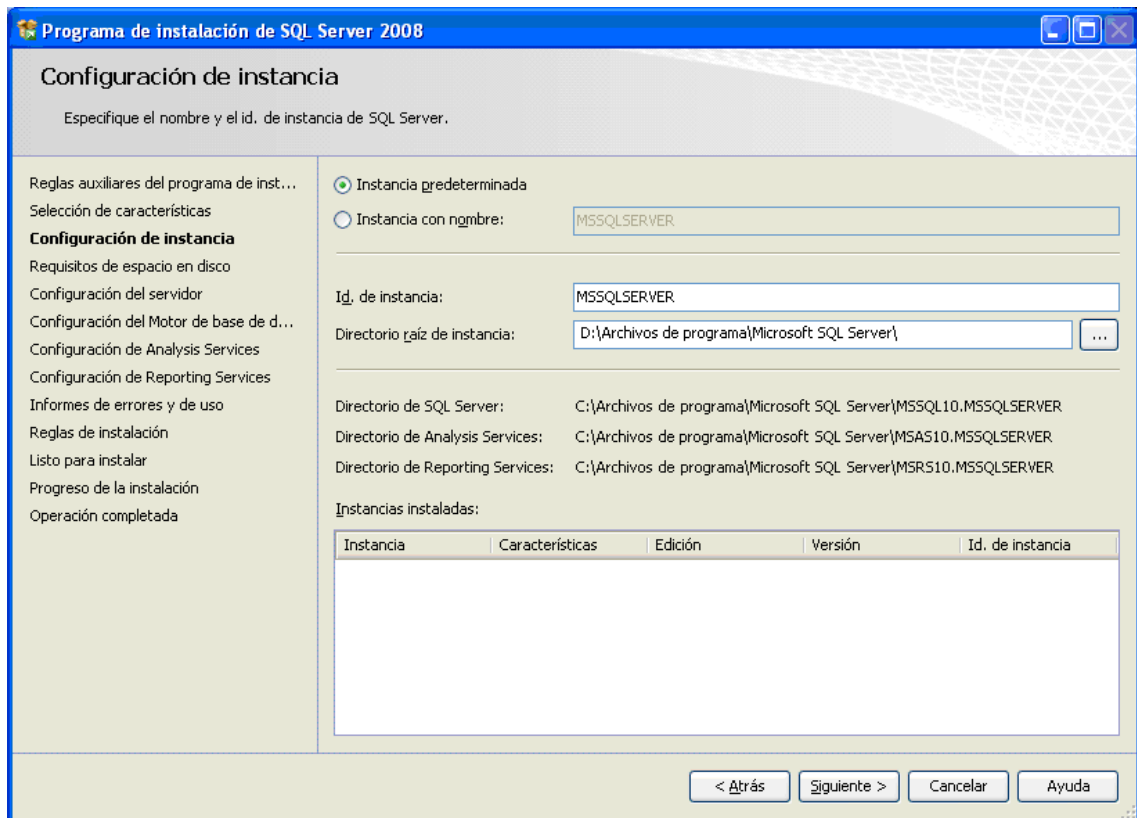


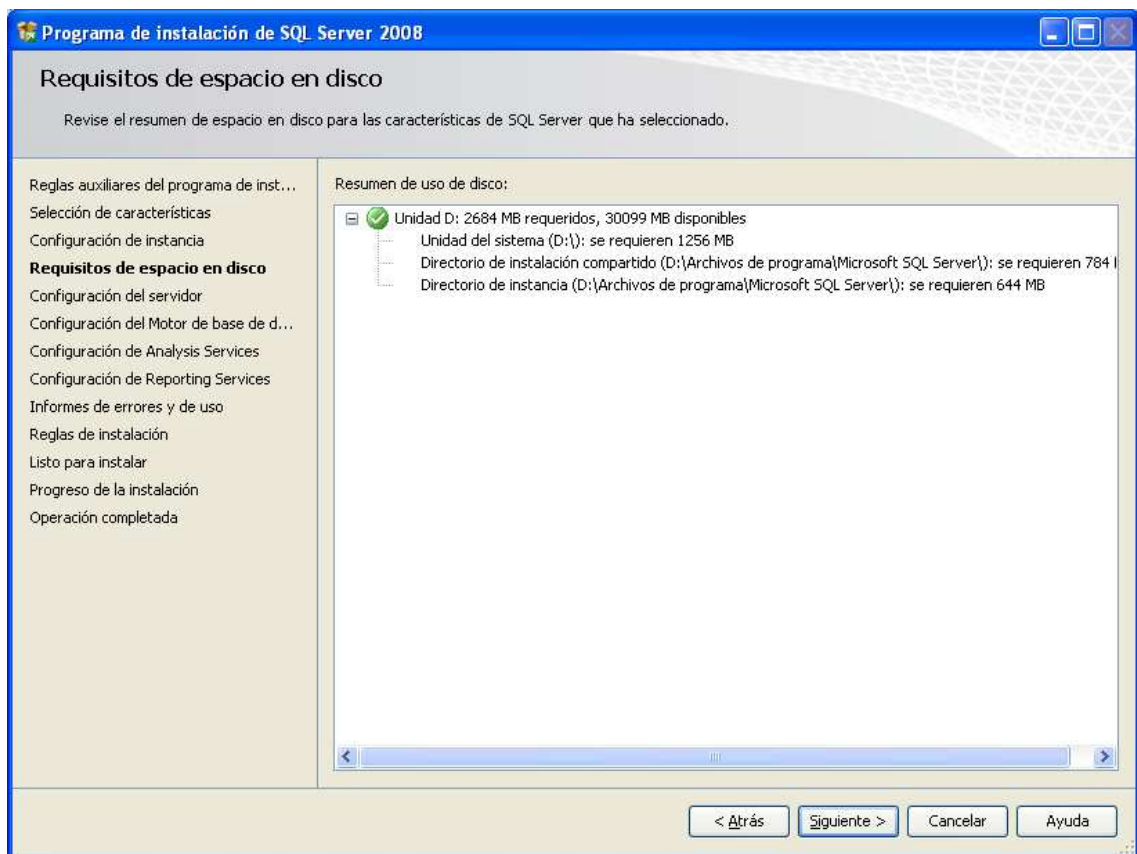
Ilustración 47 Programa de instalación de SQL Server 2008 5/21

En la pantalla de la Ilustración 48 marcaremos “Instancia predeterminada” y así creamos la instancia “MSSQLSERVER” que es almacenada en el directorio “D:\Archivos de programa\Microsoft SQL Server”



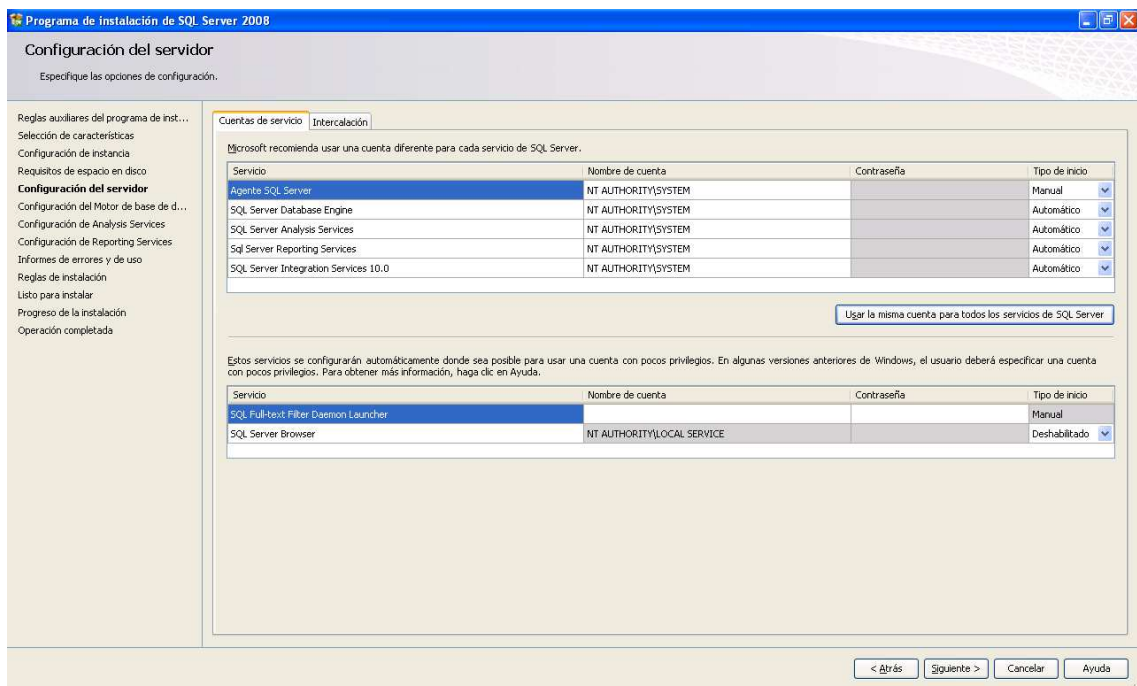
**Ilustración 48 Programa de instalación de SQL Server 2008 6/21**

En la pantalla que muestra la Ilustración 49 se comprueba y muestran los requisitos de espacio en disco para la instalación de SQL Server 2008



**Ilustración 49 Programa de instalación de SQL Server 2008 7/21**

En la pantalla de la Ilustración 50 nos da la opción de configuración de las cuentas del servidor.



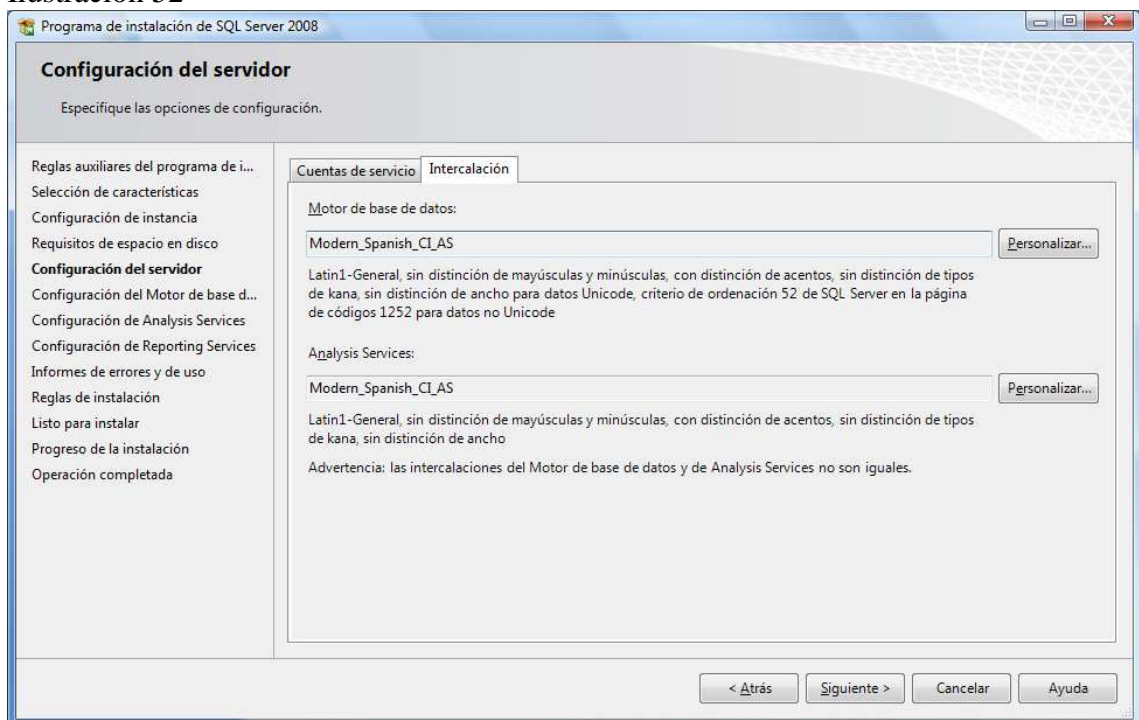
**Ilustración 50 Programa de instalación de SQL Server 2008 8/21**

Pulsamos “Usar la misma cuenta para todos los servicios de SQL Server” que nos lleva a la Ilustración 51. Donde seleccionamos “NT AUTHORITY\SYSTEM” que hace que los servicios de SQL Server se validen con una cuenta local del sistema.



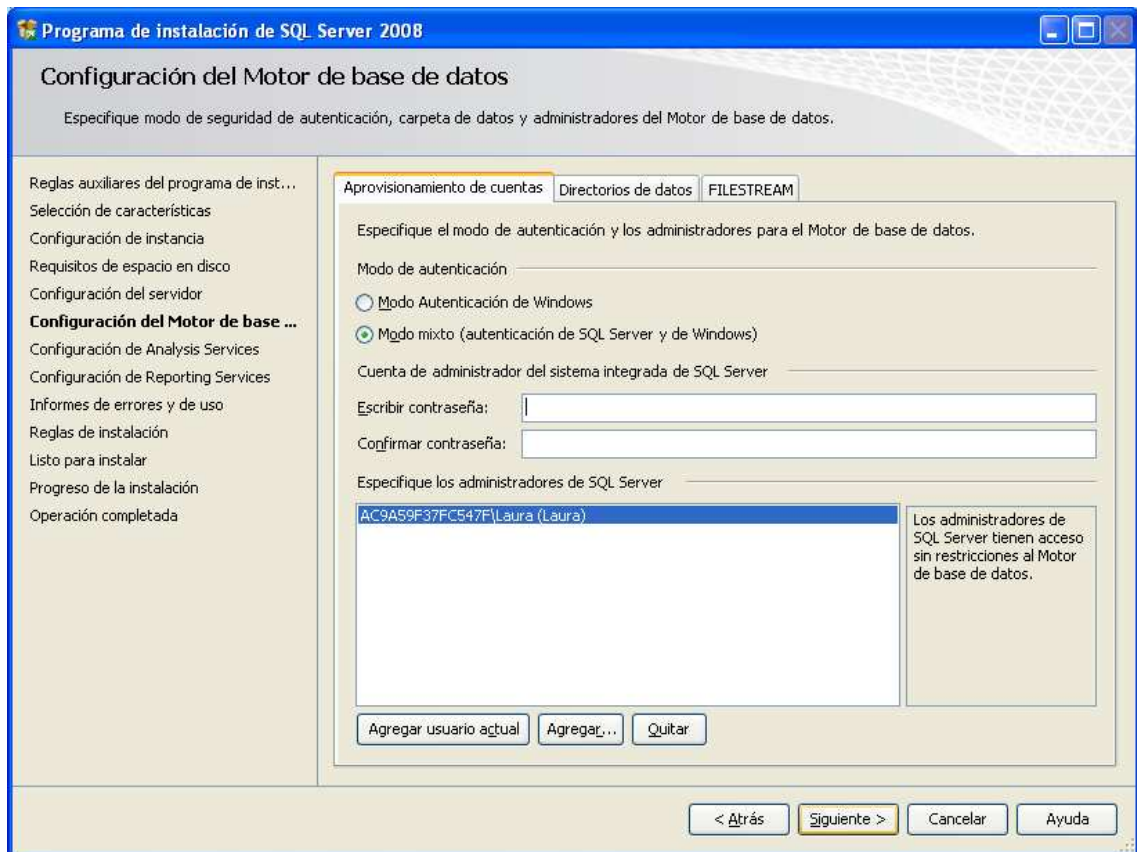
**Ilustración 51 Usar la misma cuenta para todos los servicios de SQL Server 2008**

El resto de pestañas de la pantalla permanecen como viene por defecto. Véase en la Ilustración 52



**Ilustración 52 Programa de instalación de SQL Server 2008 9/21**

En la pantalla de la Ilustración 53 se selecciona como modo de autenticación el modo mixto.

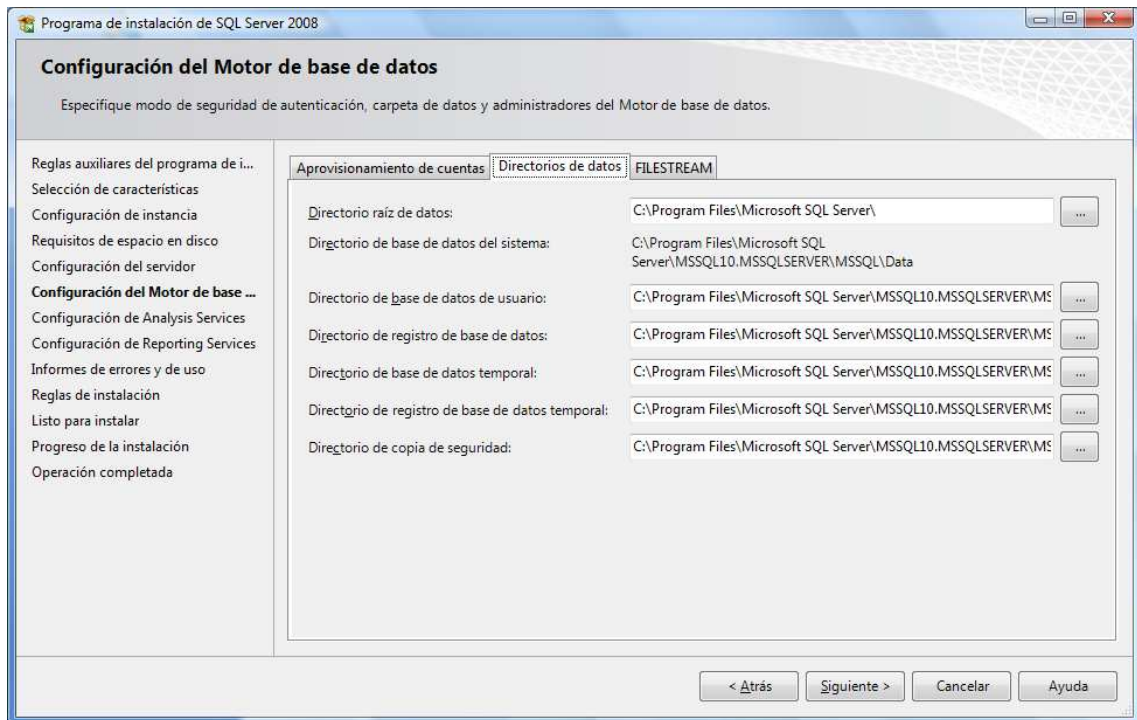


**Ilustración 53 Programa de instalación de SQL Server 2008 10/21**

Pulsamos “Agregar usuario actual” y se nos añade en usuario actual de Windows. Y le añadimos una contraseña

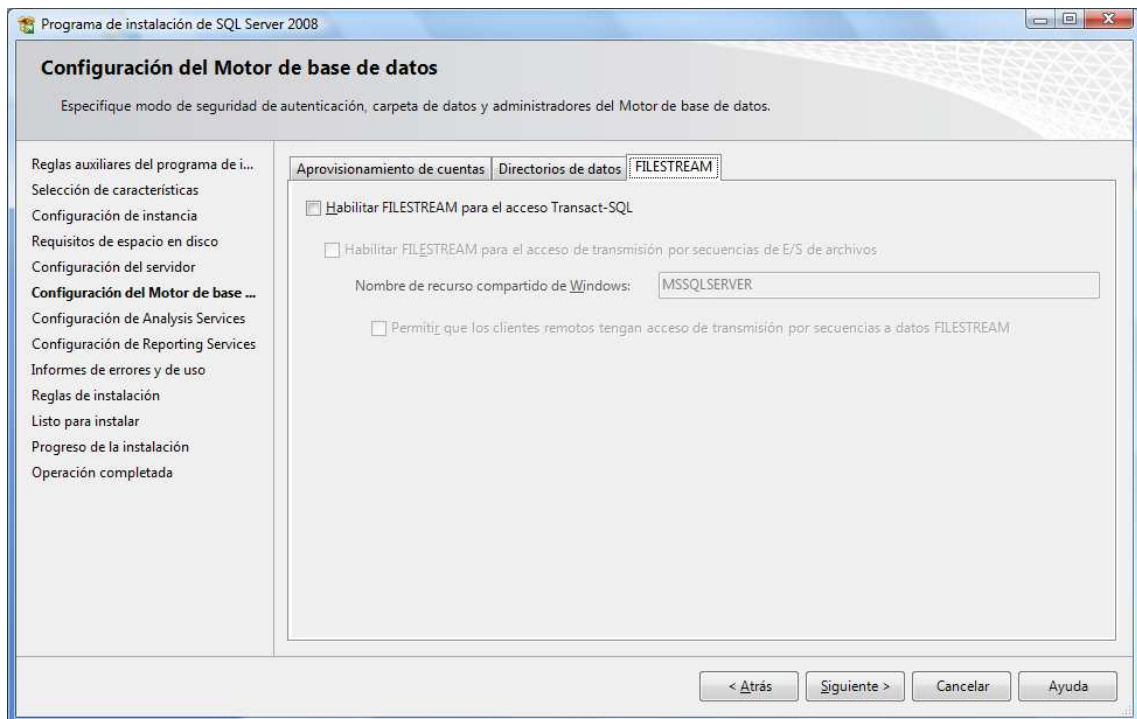
La pestaña “directorios de datos” queda como viene por defecto. Véase la Ilustración 54





**Ilustración 54 Programa de instalación de SQL Server 2008 11/21**

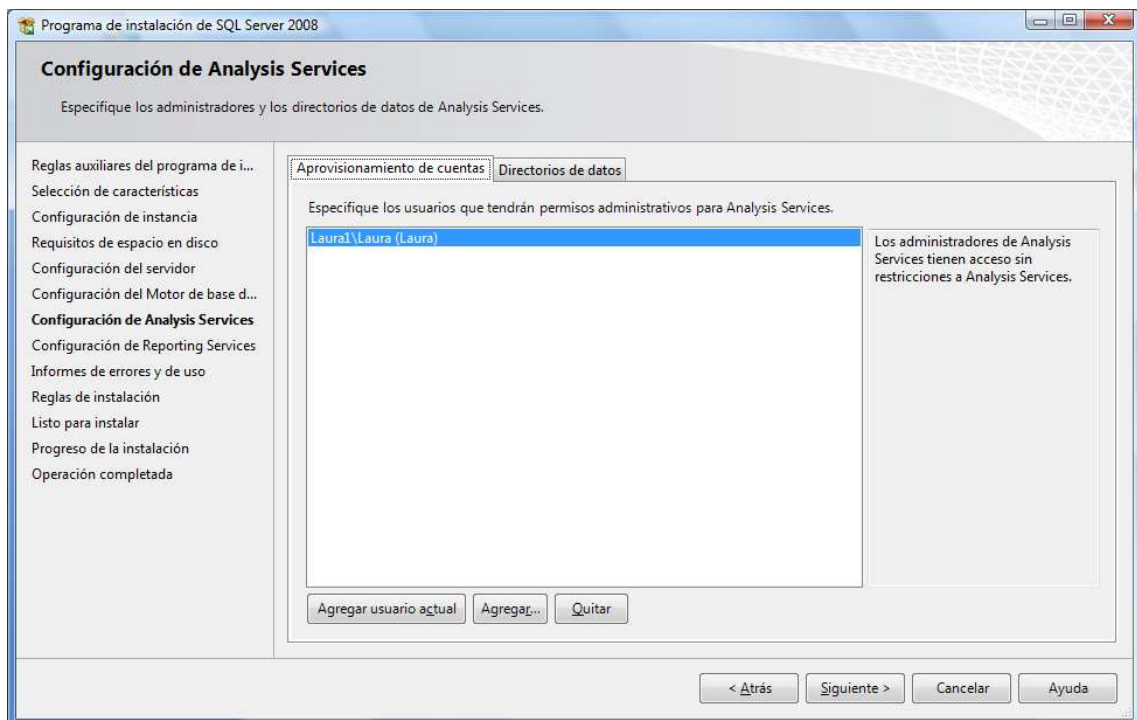
La pestaña “FILESTREAM” queda como viene por defecto. Véase la Ilustración 55



**Ilustración 55 Programa de instalación de SQL Server 2008 12/21**

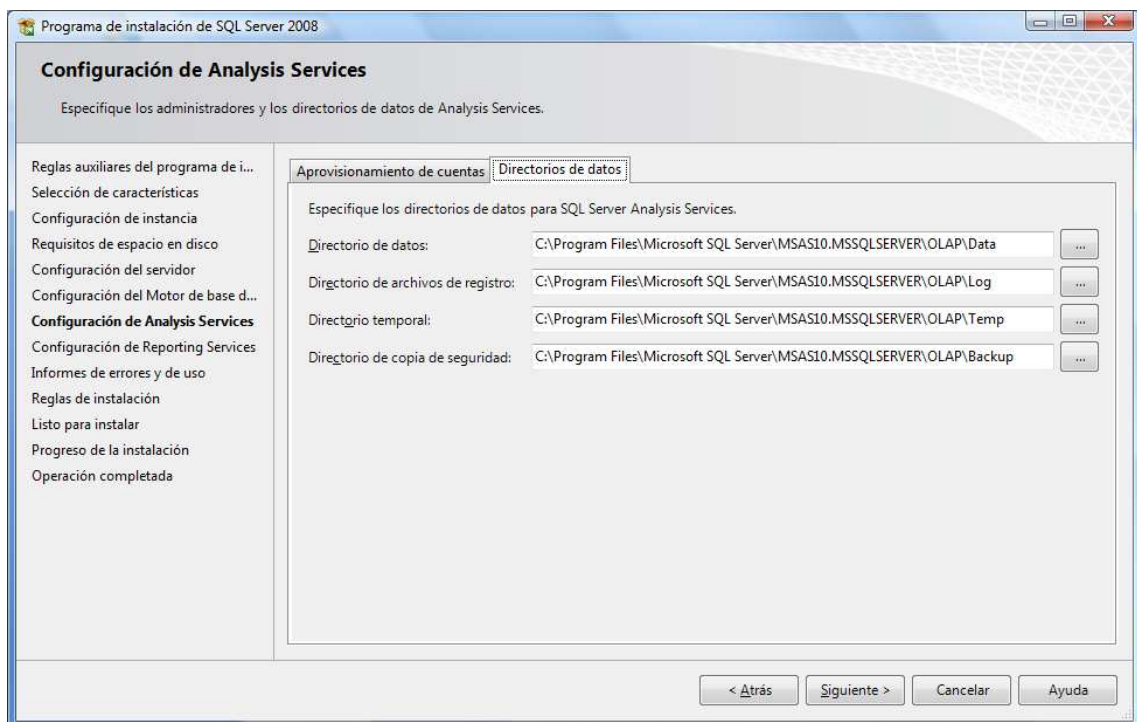
En la siguiente pantalla, que se puede ver en la Ilustración 56 al pulsar “Agregar usuario actual” se nos incluye el usuario con el que estamos conectados al sistema. Este usuario tendrá permisos administrativos para “Analysis Services”. Esta herramienta es instalada por defecto y permite analizar grandes cantidades de datos. Puede usarse para diseñar, crear y administrar estructuras multidimensionales que contengan datos de detalles y agregados de varios orígenes de datos. [6]





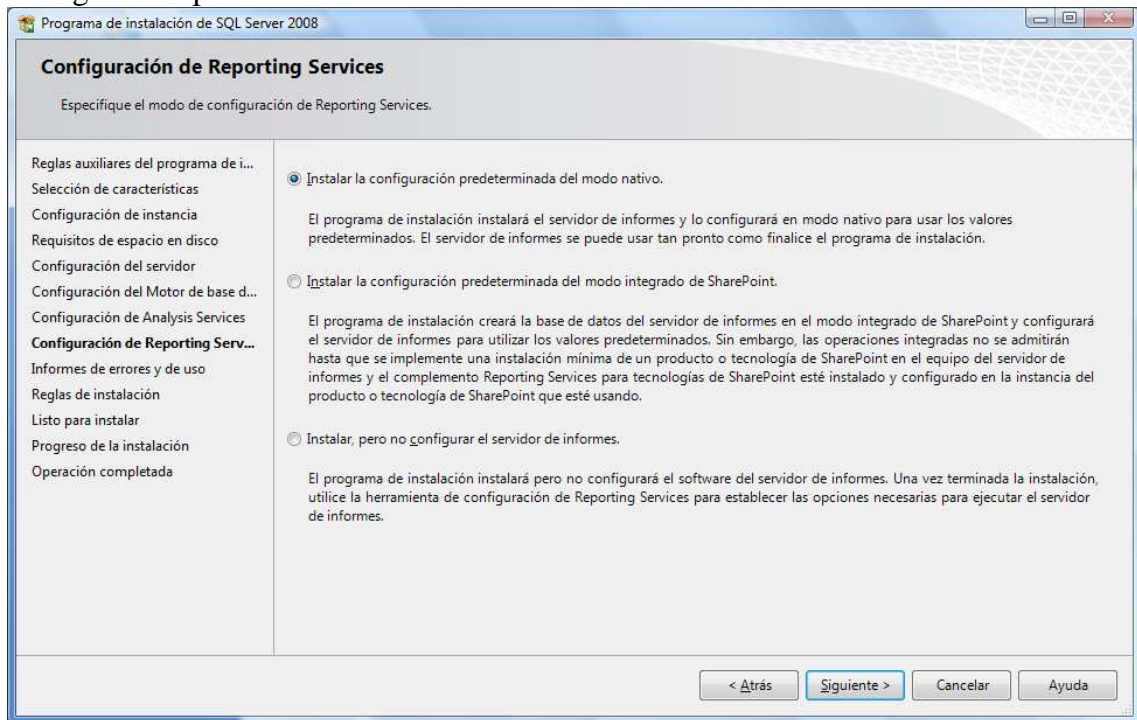
**Ilustración 56 Programa de instalación de SQL Server 2008 13/21**

La pestaña “Directorios de datos” queda como viene por defecto. Véase la Ilustración 57



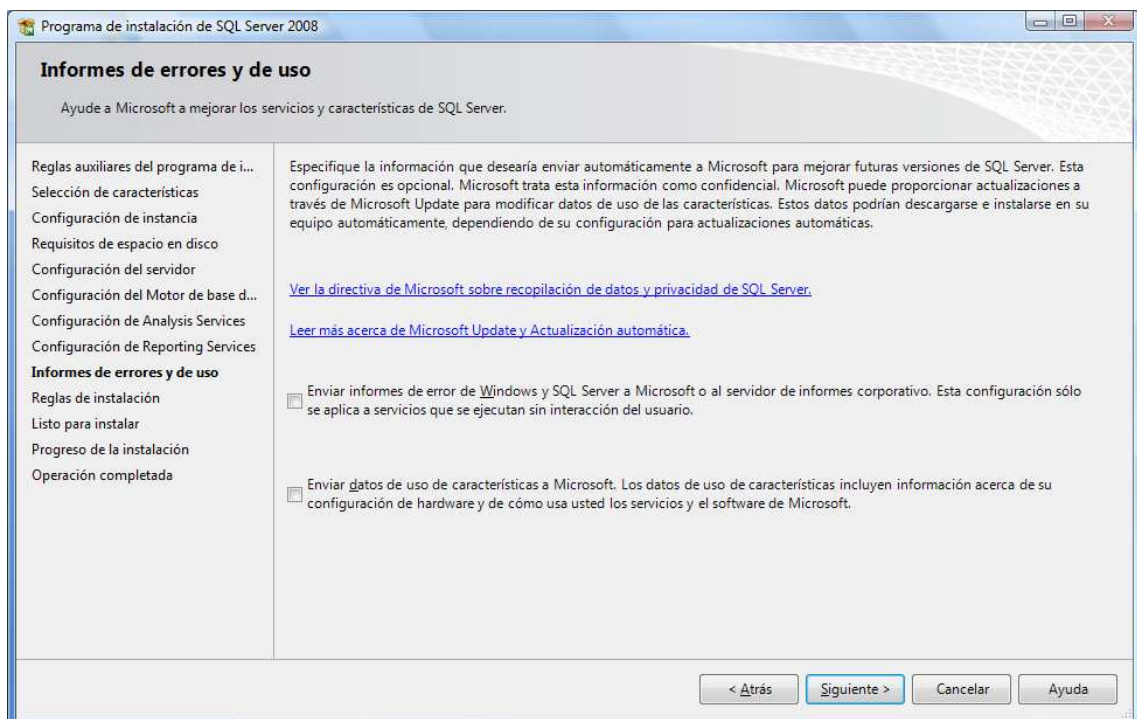
**Ilustración 57 Programa de instalación de SQL Server 2008 14/21**

En la siguiente pantalla que es mostrada por la Ilustración 58 se selecciona “Instalar configuración predeterminada del modo nativo”



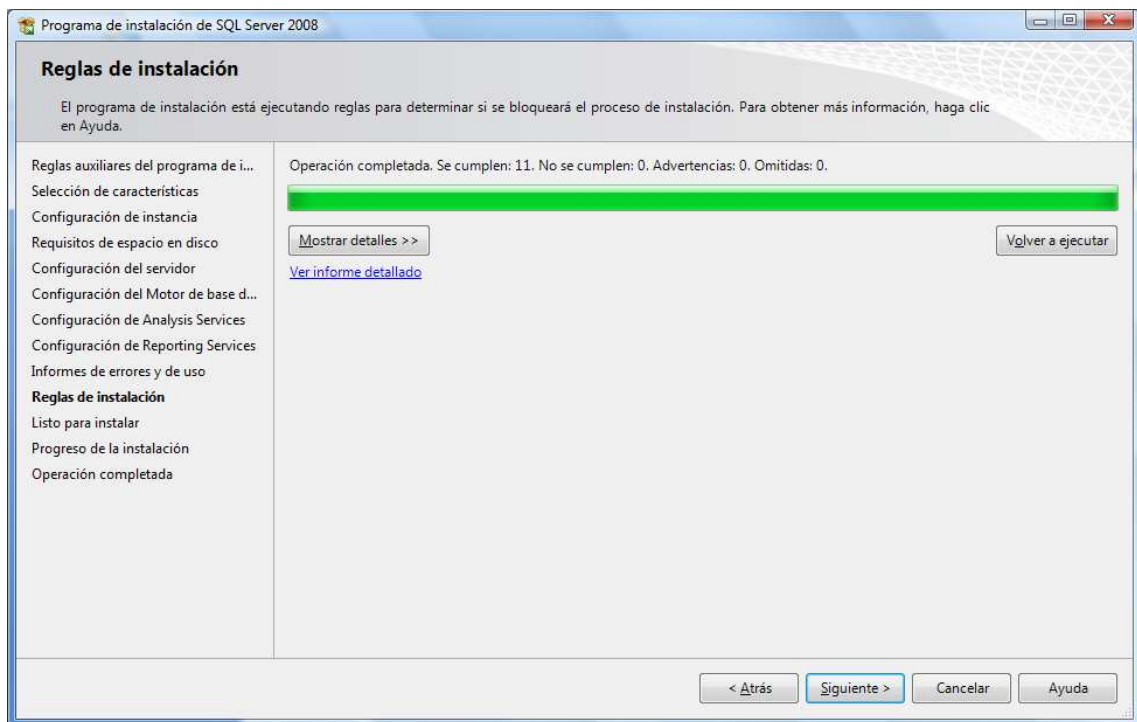
**Ilustración 58 Programa de instalación de SQL Server 2008 15/21**

En la siguiente pantalla que es mostrada por la Ilustración 59 no se seleccionan ninguna de las opciones sugeridas.



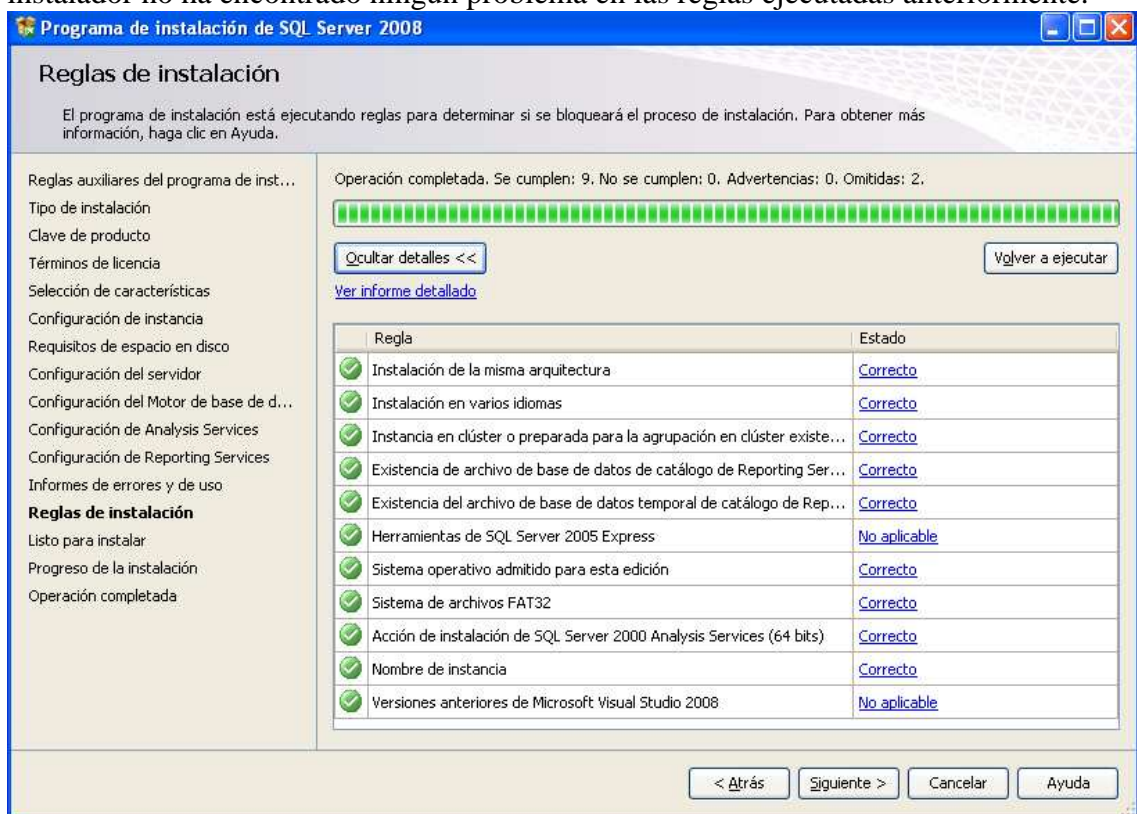
**Ilustración 59 Programa de instalación de SQL Server 2008 16/21**

En la siguiente pantalla que es mostrada por la Ilustración 60 se observa como el instalador ejecuta una serie de reglas para asegurarse que el proceso no se bloquee.



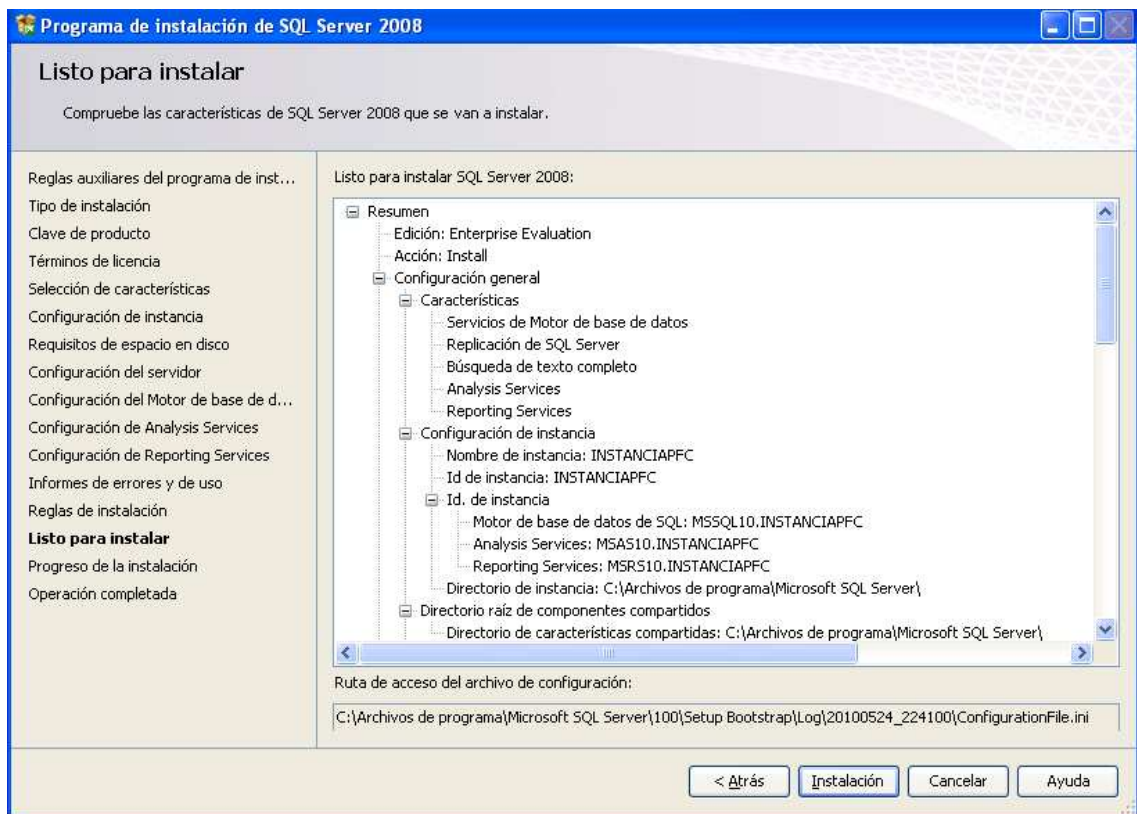
**Ilustración 60 Programa de instalación de SQL Server 2008 17/21**

En la siguiente pantalla que es mostrada por la Ilustración 61 se observa como el instalador no ha encontrado ningún problema en las reglas ejecutadas anteriormente.



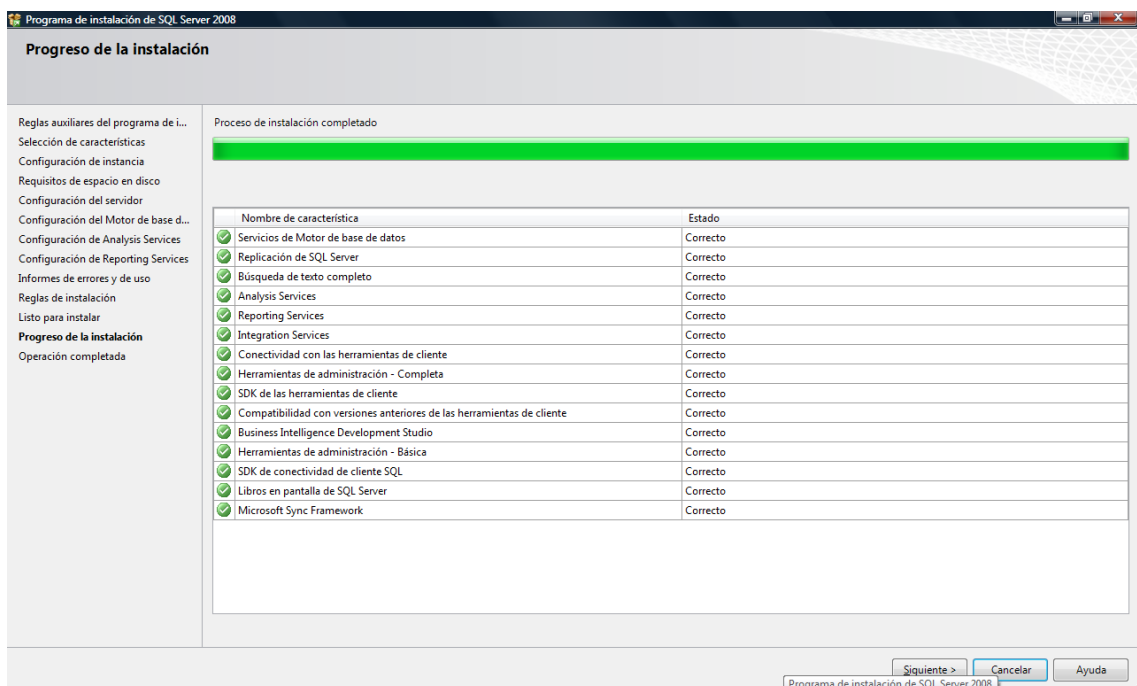
**Ilustración 61 Programa de instalación de SQL Server 2008 18/21**

No hay errores y podemos continuar, como observamos en la Ilustración 62 se listan las características de SQL Server 2008 que se van a instalar.



**Ilustración 62 Programa de instalación de SQL Server 2008 19/21**

Este proceso dura aproximadamente 80 minutos. Y en la Ilustración 63 se observa que el proceso no ha sufrido ningún problema.



**Ilustración 63 Programa de instalación de SQL Server 2008 20/21**

En la siguiente pantalla que es mostrada por la Ilustración 61 se informa de que el proceso de instalación de SQL Server se ha completado correctamente y de la ruta donde se ha almacenado el archivo de resumen de la instalación.



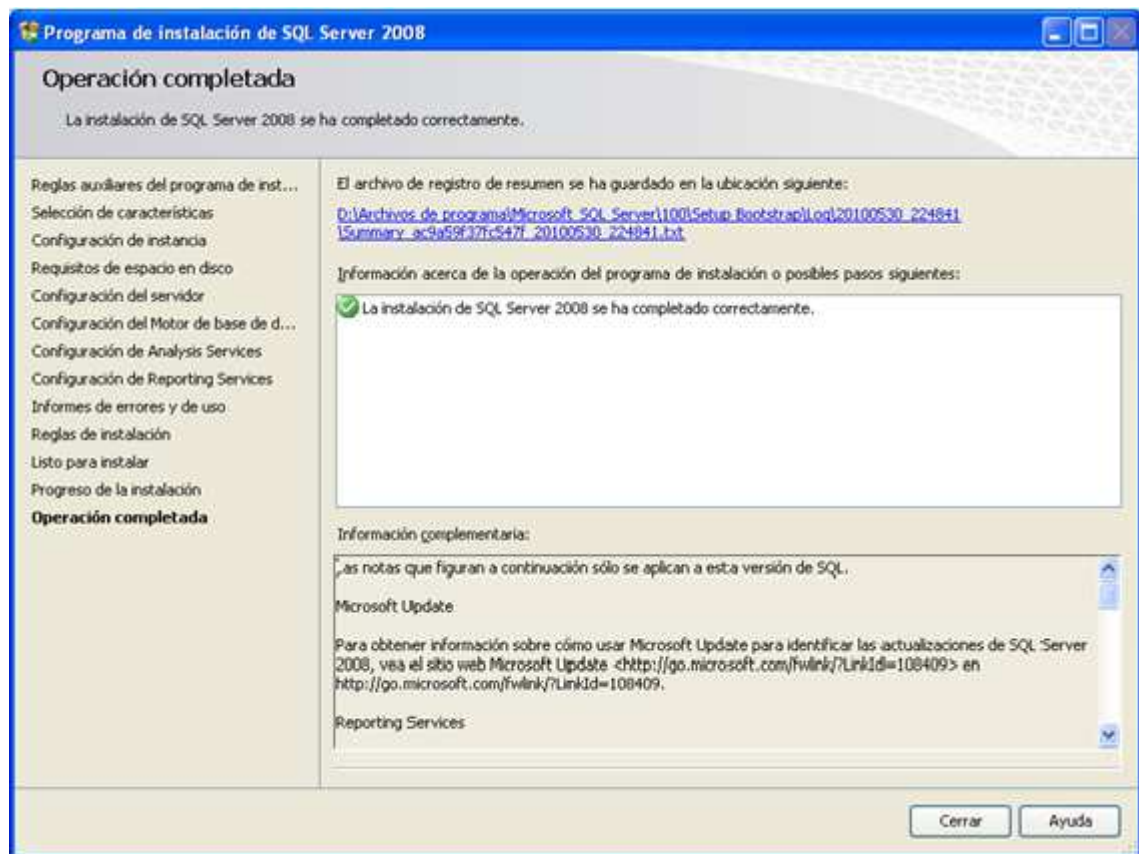


Ilustración 64 Programa de instalación de SQL Server 2008 21/21

